



(12)发明专利

(10)授权公告号 CN 104699602 B

(45)授权公告日 2017.06.23

(21)申请号 201410710567.8

(22)申请日 2014.11.28

(65)同一申请的已公布的文献号

申请公布号 CN 104699602 A

(43)申请公布日 2015.06.10

(30)优先权数据

2013-252681 2013.12.06 JP

(73)专利权人 国际商业机器公司

地址 美国纽约

(72)发明人 镰田惠介

(74)专利代理机构 北京市中咨律师事务所

11247

代理人 宛丽宏 于静

(51)Int.Cl.

G06F 11/36(2006.01)

G06F 9/44(2006.01)

(56)对比文件

CN 101604290 A,2009.12.16,

CN 102511037 A,2012.06.20,

CN 101989303 A,2011.03.23,

CN 101178687 A,2008.05.14,

CN 102880547 A,2013.01.16,

审查员 刘冰瑶

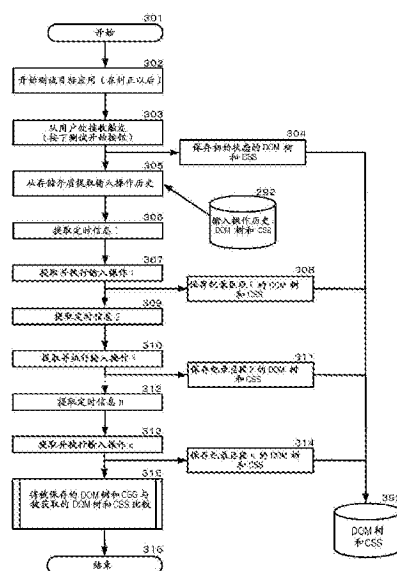
权利要求书3页 说明书25页 附图18页

(54)发明名称

用于检测影响的方法和计算机

(57)摘要

本发明具有使得由改变应用的源代码而引起的影响能够被自动检测的目标。本发明提供用于检测影响的技术,并且此技术包含除了保存用于当期待检查屏幕状态时的至少一个定时中的每一个定时的DOM树和CSS之外,还保存用户在应用上的输入操作,在源代码被改变以后,按照操作的顺序仿真输入操作,并且获取用于与当期待检查屏幕状态时的至少一个定时中的每一个定时相同的定时的DOM树和CSS,对该相同的定时中的每一个,将被保存的DOM树和CSS与被获取的DOM树和CSS比较,以及输出比较的结果。



1. 一种用于检测由改变从中可提取文档对象模型DOM树和层叠样式表CSS的应用的源代码而引起的引起的方法,所述方法引起计算机执行以下步骤:

保存用户在所述应用上的输入操作、和用于当期待检查屏幕状态时的至少一个定时中的每一个定时的DOM树和CSS;

所述源代码被改变以后,按照操作的顺序仿真所述输入操作,并且获取用于与当屏幕状态期待被检查时的至少一个定时的每一个定时相同的定时的DOM树和CSS;

针对所述相同的定时中的每一个,将在所述保存步骤中被保存的DOM树和CSS与在所述仿真步骤中被获取的DOM树和CSS比较;以及

输出所述比较的结果。

2. 根据权利要求1所述的方法,其中所述输出步骤包括以下显示步骤:在所述相同的定时时,显示CSS文件和所述CSS文件的位置,在该位置处在所述应用上的图形用户接口之间的不同的部分被定义。

3. 根据权利要求1所述的方法,其中所述比较步骤包括以下步骤:

在已经被保存的CSS或者已经被获取的CSS的CSS特性中规定第一CSS特性,所述第一CSS特性用于当期待检查屏幕状态时的至少一个时机中的一个特定定时时的实际的屏幕显示;

在已经被获取的CSS的CSS特性中规定第二CSS特性,所述第二CSS特性用于与所述一个特定定时相同的定时时的实际的屏幕显示;以及

将所述第一CSS特性与所述第二CSS特性比较。

4. 根据权利要求3所述的方法,其中所述输出步骤包括输出将所述第一CSS特性与所述第二CSS特性比较的结果的步骤。

5. 根据权利要求3所述的方法,其中所述输出步骤包括以下显示步骤:以突出的方式显示在所述已经被保存的CSS中的CSS特性中与所述第一CSS特性对应的CSS特性,所述第一CSS特性定义在所述一个特定定时时被用于实际的屏幕显示的值,并且所述与所述第一CSS特性对应的CSS特性定义没有被用于实际的屏幕显示的值。

6. 根据权利要求3所述的方法,其中所述输出步骤包括以下显示步骤:以突出的方式显示在所述已经被获取的CSS中的CSS特性中与所述第二CSS特性对应的CSS特性,所述第二CSS特性定义在与所述一个特定定时相同的定时时被用于实际的屏幕显示的值,并且所述与所述第二CSS特性对应的CSS特性定义没有被用于实际的屏幕显示的值。

7. 根据权利要求1所述的方法,其中所述保存步骤包括以下步骤:

针对已经被保存的DOM树中的每一个DOM元素,保存将应用到DOM元素上的CSS特性以及所述CSS特性的定义位置,或

针对已经被保存的DOM树中的每一个DOM元素,在应用到DOM元素上的CSS特性未定义的情况下,获取被当作初始值而使用的值,以及

所述仿真步骤包括以下步骤:

针对已经被获取的DOM树中的每一个DOM元素,获取将应用到DOM元素上的CSS特性以及所述CSS特性的定义位置,或

针对已经被获取的DOM树中的每一个DOM元素,在应用到DOM元素上的CSS特性未定义的情况下,获取被当作初始值而使用的值。

8. 根据权利要求7所述的方法, 其中所述输出步骤包括显示所述已经被保存的DOM树的步骤, 其中被显示的DOM树与被获取的DOM树之间的不同的DOM元素被突出。

9. 根据权利要求8所述的方法, 其中所述输出步骤包括以下显示的步骤: 响应于对在所述显示步骤中被突出的DOM元素的选择, 显示将应用到所述被选择的DOM元素上的CSS特性以及所述CSS特性的定义位置。

10. 根据权利要求9所述的方法, 其中所述输出步骤包括显示所述被获取的DOM树的步骤, 其中被显示的DOM树与被获取的DOM树之间的不同的DOM元素被突出。

11. 根据权利要求10所述的方法, 其中所述输出步骤包括以下显示的步骤: 响应于对所述被突出的DOM元素的选择, 显示将应用到所述被选择的DOM元素上的CSS特性以及所述CSS特性的定义位置。

12. 根据权利要求1所述的方法, 其中所述输出步骤包括输出在所述相同的定时时所述应用上的图形用户接口之间的不同的部分的步骤。

13. 根据权利要求1所述的方法, 其中

所述保存步骤进一步包括针对当期待检查屏幕状态时的至少一个定时中的每一个, 捕捉所述应用上的屏幕快照的步骤,

所述仿真步骤进一步包括在与当期待检查屏幕状态时的至少一个定时的每一个定时相同的定时时, 捕捉所述应用上的屏幕快照的步骤, 以及

所述输出步骤包括以下显示步骤: 响应于根据所述比较的结果的差异, 显示在所述保存步骤中被捕捉的屏幕快照和在所述仿真步骤中被获取的屏幕快照中的每一个。

14. 根据权利要求13所述的方法, 其中所述输出步骤包括响应于根据所述比较的结果的差异而突出所述差异的步骤, 所述差异是在所述保存步骤中被捕捉的屏幕快照和所述仿真步骤中被获取的屏幕快照之间的差异。

15. 根据权利要求1所述的方法, 其中所述输出步骤包括显示在不可见的区域中的差异的步骤。

16. 根据权利要求1所述的方法, 进一步引起所述计算机执行以下步骤:

所述源代码被进一步改变以后, 按照操作的顺序仿真所述输入操作, 并且获取用于与当期待检查屏幕状态时的至少一个定时中的每一个定时相同的定时的DOM树和CSS;

针对所述相同的定时中的每一个, 将所述源代码被改变以后被保存的DOM树和CSS与所述源代码被进一步改变以后被保存的DOM树和CSS比较; 以及

针对所述相同的定时, 输出将所述源代码被改变以后被保存的DOM树和CSS与所述源代码被进一步改变以后被保存的DOM树和CSS比较的结果。

17. 一种用于检测由改变从中可提取文档对象模型DOM树和层叠样式表CSS的应用的源代码而引起的影响的方法, 所述方法引起计算机执行以下步骤, 该计算机能够访问存储介质, 在该存储介质中保存用户在所述应用上的输入操作历史以及保存当期待检查屏幕状态时的至少一个定时时的DOM树和CSS:

所述源代码被改变以后, 按照操作的顺序仿真所述输入操作, 并且获取用于与当期待检查屏幕状态时的至少一个定时中的每一个定时相同的定时的DOM树和CSS;

针对所述相同的定时中的每一个, 将被保存在所述存储介质中的DOM树和CSS与在所述仿真步骤中被获取的DOM树和CSS比较; 以及

输出所述比较的结果。

18. 一种用于检测由改变从中可提取文档对象模型DOM树和层叠样式表CSS的应用的源代码而引起的影响的计算机,所述计算机包括:

保存部件,其用于保存用户在所述应用上的输入操作、和用于当期待检查屏幕状态时的至少一个定时中的每一个定时的DOM树和CSS;

获取部件,其用于在所述源代码被改变以后,按照操作的顺序仿真所述输入操作,并且获取用于与当期待检查屏幕状态时的至少一个定时中的每一个定时相同的定时的DOM树和CSS;

比较部件,其用于对所述相同的定时的每一个,将通过所述保存部件被保存的DOM树和CSS与通过所述仿真部件被获取的DOM树和CSS比较;以及

输出部件,其用于输出所述比较的结果。

19. 一种用于检测由改变从中可提取文档对象模型DOM树和层叠样式表CSS的应用的源代码而引起的影响的计算机,该计算机能够访问存储介质,在该存储介质中保存用户在所述应用上的输入操作历史以及保存当期待检查屏幕状态时的至少一个定时的DOM树和CSS,所述计算机包括:

获取部件,其用于在所述源代码被改变以后,按照操作的顺序仿真所述输入操作,并且获取用于与当期待检查屏幕状态时的至少一个定时中的每一个定时相同的定时的DOM树和CSS;

比较部件,其用于针对所述相同的定时的每一个,将被保存在所述存储介质中的DOM树和CSS与通过所述仿真部件被获取的DOM树和CSS比较;以及

输出部件,其用于输出所述比较的结果。

用于检测影响的方法和计算机

技术领域

[0001] 本发明关于用于检测由改变应用的源代码而引起的影响的技术,文档对象模型(“Document Object Model”,下文中也将其称为DOM)树和层叠样式表(“Cascading Style Sheets”,下文中也将其称为CSS)可以从该应用提取。

背景技术

[0002] 通过网络应用(web application)显示的屏幕可以通过改变该网络应用的源代码而改变(例如,通过纠正缺陷、或通过改变规范)。所以,当网络应用的源代码被改变时,必须检查回归或退化是否已经发生。正常地,对此检查,每次源代码被改变时测试者在操作网络浏览器时执行视觉回归测试。

[0003] 但是,通过回归测试手工地执行每一个测试是代价极其高的。于是,正常地,回归测试只对一部分执行,该部分被假定受到网络应用的源代码中的改变的影响。但是用这样的回归测试,归因于寻找可能受影响的所有部分的失败,回归或退化经常发生。

[0004] 而且,例如在可以影响整个网络应用的公共JavaScript(注册商标)框架被改变的情况下,必须对所有的屏幕执行回归测试。但是,如果网络页面是通过使用上文提到的JavaScript(注册商标)框架层次化的并且是动态地被创建的,对由改变源代码而引起的网络页面上的影响的范围的调查是极其困难的。而且,因为在网络应用上的回归测试是一次又一次地重复相似的操作的工作,所以对于测试者来说这是极其枯燥的工作。

[0005] 于是,用于自动化在网络应用上的回归测试的多个软件被提供。例如,这些软件是开源软件Selenium(见下文提到的非专利文献1和非专利文献2至4)和以下商业产品、**IBM® Rational®** Functional Tester(国际商业机器公司)(见下文提到的非专利文献5和6)、SmartBear TestComplete(见下文提到的非专利文献7)和**HP®** QuickTest Professional(见下文提到的非专利文献8)。上文提到的Selenium、**IBM® Rational®** Functional Tester、SmartBear TestComplete和**HP®** QuickTest Professional所有都采用用于在清楚地指示对比部分时执行回归测试的白名单方法。

[0006] 使用Selenium,通过创建用于网络应用的测试用例,回归测试几乎可以完全地自动执行。而且,因为Selenium使得在多个网络浏览器上能够容易地执行回归测试,所以可以增加在网络应用上的回归测试的效率。

[0007] 下文提到的专利文献1描述了用于应用程序的测试程序,并且此测试程序引起安装应用程序的计算机起到记录控制部件、执行部件和错误输出控制部件的功能,该记录控制部件用于在记录装置中累积地记录在测试用例于该应用程序上执行时由该应用程序生成的用户事件和屏幕事件,该执行部件用于读取累积地记录在记录装置中的用户事件和屏幕事件并且通过作为测试目标的应用程序连续地执行用户事件,该错误输出控制部件用于将从由作为测试目标的应用程序进行的一个用户事件的执行到下一个用户事件的执行而生成的屏幕事件,与从自存储装置中读取出的一个用户事件的执行到下一个用户事件的执

行而生成屏幕事件相比较,以及用于输出不相符的屏幕事件作为错误(权利要求1)。进一步地,专利文献1描述通过记录控制部件被累积地记录在记录装置中的屏幕事件是屏幕改变的差异,并且是要通过文档对象模型(DOM)获取的信息(权利要求4),以及通过记录控制部件被累积地记录在记录装置中的屏幕事件是由通过Xpath和DOM子树表达的改变部分所指示的信息(权利要求5)。

[0008] 下文提到的专利文献2描述回归测试支持程序,并且该回归测试支持程序引起计算机执行以下过程:检测一个或多个被显示在显示器上的浏览器屏幕中的一个的显示内容的更新,当每次检测到更新时在第一存储单元中记录数据(该数据包含指示所有被显示的浏览器屏幕的显示内容的信息和关于该更新的浏览器屏幕的识别信息),根据对数据的编辑指令引用第一存储单元,以及显示图像的清单,这些图像指示与识别信息关联的浏览器屏幕的显示内容(权利要求1)。

[0009] 【引用文献列表】

[0010] 【专利文献】

[0011] 【专利文献1】JP2009-140155A

[0012] 【专利文献2】JP2013-54693A

[0013] 【非专利文献】

[0014] 【非专利文献1】“Selenium HQ”,(在网络上),(于2013年9月26日检索到),互联网<URL:<http://docs.seleniumhq.org/>>

[0015] 【非专利文献2】Takashi OKAMOTO,“Selenium for automating Web test regardless of browser(1/3)”,(在网络上),(于2013年9月27日检索到),互联网

[0016] <<http://www.atmarkit.co.jp/ait/articles/0908/19/news109.html>>

[0017] 【非专利文献3】Takashi OKAMOTO,“Selenium for automating Web test regardless of browser(2/3)”,(在网络上),(于2013年9月27日检索到),互联网

[0018] <http://www.atmarkit.co.jp/ait/articles/0908/19/news109_2.html>

[0019] 【非专利文献4】Takashi OKAMOTO,“Selenium for automating Web test regardless of browser(3/3)”,(在网络上),(于2013年9月27日检索到),互联网

[0020] <http://www.atmarkit.co.jp/ait/articles/0908/19/news109_3.html>

[0021] 【非专利文献5】“Rational Functional Tester(Functional Test/Regression Test)”,(在网络上),(于2013年9月27日检索到),互联网<URL:

[0022] <http://www-06.ibm.com/software/jp/rational/products/test/rft/>>

[0023] 【非专利文献6】Kazunari HOHSHOH等,“Effective System Development Practice Using Tools-Aiming at Smarter System Development-”,IBM Japan, PROVISION Spring No.69,2011,从互联网可以得到<URL:

[0024] http://www-06.ibm.com/ibm/jp/provision/no69/pdf/69_article3.pdf>

[0025] 【非专利文献7】“SMARTBEAR TestComplete,Reducing Cost by Automation of GUI Testing”,(在网络上),(于2013年9月27日检索到),互联网<URL:

[0026] <http://www.xlsoft.com/jp/products/smartbear/testcomplete.html?tab=0>>

[0027] 【非专利文献8】“HP QuickTest Professional 10.0 software”,(在网络上),(于2013年9月27日检索到),互联网<URL:

[0028] <http://h50146.www5.hp.com/products/software/hpsoftware/solutions/lineup/qtp/new.html>>

发明内容

[0029] 【技术问题】

[0030] 对于用于应用的回归测试、用于应用的单元测试、或用于构建应用的代码研究,对影响的检测是重要的,该影响由改变任意应用的源代码而引起,从该应用可以提取DOM树和CSS。于是,本发明具有使得由改变应用的源代码而引起的影响能够被自动检测的目标。

[0031] 而且,要检查应用与多种网络浏览器或同一网络浏览器的不同版本兼容,必须执行“浏览器数量×版本数量”次数的测试,并且这是成本极其高的。进一步地,即使网络浏览器是相同的,如果对例如**Windows®**、**Mac OS®** X、**Linux®**、**Android®**和iOS的操作系统的每一个以及对每一个支持的版本要检查应用的行为(包含屏幕显示和除了屏幕显示以外的不可见区域的行为),代价将会进一步增加。于是,要减少代价,本发明具有它的目标,是要使得能够自动地检测由改变应用的源代码而引起的影响。

[0032] 进一步地,对用于在网络应用上自动化回归测试的常规软件是采用白名单方法的情况,本发明具有它的目标,是要消除单独地写被白名单方法期待的定义(例如“项目X的值应该是ABC”的定义)、测试用例或统一建模语言(UML,“unified modeling language”)的需要。

[0033] 更进一步地,本发明具有它的目标,是要使得能够检测关于设计的问题(例如行偏移量、画图失败、或重叠的项目)。具体地,以使得能够检测没有在屏幕显示上示出的差异(例如,在不可见的区域中的差异或者归因于显示大小的差异)为目标。

[0034] 更进一步地,本发明具有它的目标,是要使得关于这样的部分能够进行自动的确定,其中关于该部分的被白名单方法期待的定义是困难的,并且关于该部分依赖于用户的眼睛。

[0035] 更进一步地,本发明具有它的目标,是要使得在尽可能地减少用户的参与的同时能够检测由改变应用的源代码而引起的影响。

[0036] 【解决技术问题的技术方案】

[0037] 本发明提供用于检测由改变从中可以提取DOM树和CSS的应用的源代码而引起的影响的技术。此技术可以包含用于检测由改变应用的源代码而引起的影响的方法、用于检测该影响的计算机、计算机程序以及计算机程序产品。

[0038] 根据本发明的第一方面,用于检测由改变从中可以提取DOM树和CSS的任意应用的源代码而引起的影响的方法引起计算机执行以下步骤:

[0039] 保存用户在应用上的输入操作、和用于当期待检查屏幕状态时的至少一个定时中的每一个定时的DOM树和CSS;

[0040] 源代码被改变以后,按照操作的顺序仿真该输入操作,并且获取用于与当期待检查屏幕状态时的至少一个定时中的每一个定时相同的定时的DOM树和CSS;

[0041] 针对该相同的定时中的每一个,将在保存步骤中被保存的DOM树和CSS与在仿真步骤中被获取的DOM树和CSS比较;以及

[0042] 输出比较的结果。

[0043] 根据本发明的第二方面,用于检测由改变从可以提取DOM树和CSS的任意应用的源代码而引起的的影响的方法引起计算机执行以下步骤,其中该计算机能够访问存储介质,在该存储介质中保存用户在应用上的输入操作历史以及保存当期待检查屏幕状态时的至少一个定时时的DOM树和CSS:

[0044] 源代码被改变以后,按照操作的顺序仿真该输入操作,并且获取用于与当期待检查屏幕状态时的至少一个定时中的每一个定时相同的定时的DOM树和CSS;

[0045] 针对该相同的定时中的每一个,将被保存在该存储介质中的DOM树和CSS与在仿真步骤中被获取的DOM树和CSS比较;以及

[0046] 输出比较的结果。

[0047] 根据本发明的实施例,根据上文所描述的第一或第二方面的方法可以进一步引起计算机执行以下步骤:

[0048] 源代码被进一步改变以后,按照操作的顺序仿真该输入操作,并且获取用于与当期待检查屏幕状态时的至少一个定时中的每一个定时相同的定时的DOM树和CSS;

[0049] 针对该相同的定时中的每一个,将源代码被改变以后被保存的DOM树和CSS与源代码被进一步改变以后被保存的DOM树和CSS比较;以及

[0050] 针对该相同的定时,输出将源代码被改变以后被保存的DOM树和CSS与源代码被进一步改变以后被保存的DOM树和CSS比较的结果。

[0051] 根据本发明的第三方面,用于检测由改变从中可以提取DOM树和CSS的任意应用的源代码而引起的影响的计算机包括以下部件,其中:

[0052] 保存部件,其用于保存用户在该应用上的输入操作、和用于当期待检查屏幕状态时的至少一个定时中的每一个定时的DOM树和CSS;

[0053] 获取部件,其用于在源代码被改变以后,按照操作的顺序仿真该输入操作,并且获取用于与当期待检查屏幕状态时的至少一个定时中的每一个定时相同的定时的DOM树和CSS;

[0054] 比较部件,其用于针对该相同的定时中的每一个,将通过该保存部件被保存的DOM树和CSS与通过该仿真部件被获取的DOM树和CSS比较;以及

[0055] 输出部件,其用于输出比较的结果。

[0056] 根据本发明的实施例,在根据上文描述的第三方面的计算机中,针对已经被保存的DOM树中的每一个DOM元素,该保存部件可以保存将应用到DOM元素上的CSS特性以及该CSS特性的定义位置。

[0057] 根据本发明的实施例,在根据上文描述的第三方面的计算机中,对已经被获取的DOM树中的每一个DOM元素,该获取部件可以获取将应用到DOM元素上的CSS特性以及该CSS特性的定义位置。而且,在将应用到DOM元素上的CSS特性未定义的情况中,针对已经被获取的DOM树中的每一个DOM元素,该获取部件可以获取被当作初始值而使用的值。

[0058] 根据本发明的实施例,在根据上文描述的第三方面的计算机中,

[0059] 该保存部件可以针对当期待检查屏幕状态时的至少一个定时中的每一个,捕捉应用上的屏幕快照,

[0060] 该仿真部件可以在与当期待检查屏幕状态时的至少一个定时中的每一个定时相同的定时时,捕捉应用上的屏幕快照,以及

[0061] 该输出部件可以响应于根据比较的结果的差异,显示通过该保存部件被捕捉的屏幕快照和通过该仿真部件被获取的屏幕快照中的每一个。

[0062] 根据本发明的实施例,在根据上文描述的第三方面的计算机中,该输出部件可以响应于根据比较的结果的差异,用突出的方式显示该差异,该差异是在通过该保存部件被捕捉的屏幕快照和通过该仿真部件被获取的屏幕快照之间的差异。

[0063] 根据本发明的第四方面,用于检测由改变从中可以提取DOM树和CSS的任意应用的源代码而引起的影响的计算机包括以下部件,其中该计算机能够访问存储介质,在该存储介质中保存用户在应用上的输入操作历史以及保存当期待检查屏幕状态时的至少一个定时时的DOM树和CSS:

[0064] 获取部件,其用于在源代码被改变以后,按照操作的顺序仿真该输入操作,并且获取用于与当期待检查屏幕状态时的至少一个定时中的每一个定时相同的定时的DOM树和CSS;

[0065] 比较部件,其用于针对该相同的定时中的每一个,将被保存在该存储介质中的DOM树和CSS与通过该仿真部件被获取的DOM树和CSS比较;以及

[0066] 输出部件,其用于输出比较的结果。

[0067] 根据本发明的实施例,根据上文描述的第四方面的计算机能够访问的该存储介质中进一步保存,针对已经被保存的DOM树中的每一个DOM元素,将应用到DOM元素上的CSS特性以及该CSS特性的定义位置,以及

[0068] 该获取部件可以针对已经被获取的DOM树中的每一个DOM元素,从该存储介质获取将应用到DOM元素上的CSS特性以及该CSS特性的定义位置,或者在应用到DOM元素上的CSS特性未定义的情况中,针对已经被获取的DOM树中的每一个DOM元素,获取被当作初始值而使用的值。

[0069] 根据本发明的实施例,根据上文描述的第四方面的计算机能够访问的该存储介质其中进一步保存,针对当期待检查屏幕状态时的至少一个定时中的每一个,被捕捉的应用上的屏幕快照,其中

[0070] 在与当期待检查屏幕状态时的至少一个定时中的每一个定时相同的定时时,该获取部件可以捕捉应用上的屏幕快照,以及

[0071] 该输出部件可以响应于根据比较的结果的差异,显示在该保存步骤中被捕捉的屏幕快照和在该获取步骤中被获取的屏幕快照中的每一个。

[0072] 根据本发明的实施例,根据上文描述的第四方面的计算机中,该输出部件可以响应于根据比较的结果的差异,用突出的方式显示在该保存步骤中被捕捉的屏幕快照和该获取步骤中被获取的屏幕快照之间的差异。

[0073] 根据本发明的实施例,根据上文描述的第三或第四方面的计算机中,该比较部件可以:

[0074] 在已经被保存的CSS的CSS特性中规定第一CSS特性,该第一CSS特性用于当期待检查屏幕状态时的至少一个定时中的一个特定定时时的实际的屏幕显示;

[0075] 在已经被获取的CSS的CSS特性中规定第二CSS特性,所述第二CSS特性用于与该一个特定定时相同的定时时的实际的屏幕显示;以及

[0076] 将该第一CSS特性与该第二CSS特性比较。

[0077] 根据本发明的实施例,根据上文描述的第三或第四方面的计算机中,该输出部件可以在该相同的定时时显示CSS文件和该CSS文件的位置,在该位置处应用上的图形用户接口之间的不同的部分被定义。

[0078] 根据本发明的实施例,根据上文描述的第三或第四方面的计算机中,该输出部件可以输出将该第一CSS特性与该第二CSS特性比较的结果。

[0079] 根据本发明的实施例,根据上文描述的第三或第四方面的计算机中,该输出部件可以用突出的方式显示在已经被保存的CSS的那些CSS特性中的与该第一CSS特性对应的CSS特性,该第一CSS特性定义在一个特定定时时被用于实际的屏幕显示的值,并且与该第一CSS特性对应的CSS特性定义没有被用于实际的屏幕显示的值。

[0080] 根据本发明的实施例,根据上文描述的第三或第四方面的计算机中,该输出部件可以用突出的方式显示在已经被获取的CSS中的那些CSS特性中的与该第二CSS特性对应的CSS特性,该第二CSS特性定义在与一个特定定时相同的定时时被用于实际的屏幕显示的值,并且与该第二CSS特性对应的CSS特性定义没有被用于实际的屏幕显示的值。

[0081] 根据本发明的实施例,根据上文描述的第三或第四方面的计算机中,该输出部件可以显示已经被保存的DOM树,关于该DOM树的显示,该输出部件用突出的方式显示在被显示的DOM树与已经被获取的DOM树之间不同的DOM元素。

[0082] 根据本发明的实施例,根据上文描述的第三或第四方面的计算机中,

[0083] 响应于对用突出的方式显示的DOM元素的选择,该输出部件可以显示将应用到该被选择的DOM元素上的CSS特性以及该CSS特性的定义位置,以及

[0084] 可选地,该输出部件可以进一步显示已经被获取的DOM树,该输出部件用突出的方式显示在被显示的DOM树与已经被获取的DOM树之间不同的DOM元素。

[0085] 根据本发明的实施例,根据上文描述的第三或第四方面的计算机中,可选地,响应于对用突出的方式显示的DOM元素的选择,该输出部件可以进一步显示将应用到该被选择的DOM元素上的CSS特性以及该CSS特性的定义位置。

[0086] 根据本发明的实施例,根据上文描述的第三或第四方面的计算机中,该输出部件可以在该相同的定时时输出应用上的图形用户接口之间的不同的部分。

[0087] 根据本发明的实施例,根据上文描述的第三或第四方面的计算机中,该输出部件可以显示在不可见的区域中的差异。

[0088] 根据本发明的实施例,根据上文描述的第三或第四方面的计算机中,

[0089] 源代码被进一步改变以后,该获取部件可以按照操作的顺序进一步仿真该输入操作,并且获取用于与当期待检查屏幕状态时的至少一个定时中的每一个定时相同的定时的DOM树和CSS;

[0090] 针对该相同的定时中的每一个,该比较部件可以将源代码被改变以后被保存的DOM树和CSS与源代码被进一步改变以后被保存的DOM树和CSS比较;以及

[0091] 针对该相同的定时,该输出部件可以输出将源代码被改变以后被保存的DOM树和CSS与源代码被进一步改变以后被保存的DOM树和CSS比较的结果。

[0092] 根据本发明的第五方面,用于检测由改变从中可以提取DOM树和CSS的任意应用的源代码而引起的影响的计算机程序或计算机程序产品引起计算机执行根据该第一方面或该第二方面的方法的每一步骤。

[0093] 根据本发明的实施例的多个计算机程序,每一个计算机程序可以被存储在一个或多个任意的计算机可读的记录介质中,该记录介质例如能够被连接到USB、ROM、MRAM、RAM等的软盘、MO、CD-ROM、DVD、BD、硬盘装置、存储器介质。这些计算机程序可以从通过通信线路连接的另一个计算机(例如服务器计算机)下载,或者可以从另一个记录介质中复制从而保存到上文提到的记录介质中。而且,根据本发明的实施例的多个计算机程序可以被压缩或分成多个碎片,并且被保存在一个或多个记录介质中。此外,应当注意到根据本发明的实施例的计算机程序产品当然可以用多种模式被提供。根据本发明的实施例的计算机程序产品可以包含例如记录计算机程序的存储介质或用于传输计算机程序的传输介质。

[0094] 上文描述的本发明的发明内容没有引证本发明的所有必需的特点,并且应当注意到这些结构的元件的组合或子组合也可以被当做本发明。

[0095] 对本领域技术人员来说多种变形是很容易地可想象到的,这些变形例如将在本发明的多个方面中使用的计算机的硬件结构的元件与多个机器组合、向其分配功能以及执行这些功能,这是顺理成章的事。这些变形当然是本发明的想法中的概念。但是,这些结构的元件仅仅只是示例性的,而且并非所有的结构的元件是本发明必需的结构元件。

[0096] 此外,本发明还可以通过硬件、软件、或硬件和软件的组合而被实现。在安装了计算机程序的计算机上的执行是通过硬件和软件的组合的执行的典型示例。在此情况中,计算机程序控制计算机并且通过被加载到计算机的存储器上 and 被执行从而引起计算机执行根据本发明的过程。计算机程序可以通过命令群被构建,该命令群可以通过任意的语言、代码或符号被表达。这样的命令群使得计算机能够直接地执行特定的功能或者在以下两项的一个或两个的执行以后执行根据本发明的实施例的过程:(1)向另一语言、代码或描述的转换,以及(2)复制到另一介质上。

[0097] 【本发明的有益效果】

[0098] 根据本发明,由改变任意应用的源代码而引起的影响的检测可以不通过列白名单而是通过列黑名单而被执行,其中从该任意应用可以提取DOM树和CSS,其中列白名单假定项目(或元素)应当具有这个值,而黑名单检测在没有差异的前提下的差异。通过列黑名单而执行的影响的检测具有这样的优势,即不需要单独地写列白名单所要求的定义(例如“项目X的值应该是ABC”的定义)、或测试用例或统一建模语言(UML)。而且,因为不需要单独地写定义,所以具有不需要验证定义信息的准确性的优势。更多地,因为不需要单独地写测试用例或UML,所以具有消除测试用例的创建者的主观看法的优势。

[0099] 此外,根据本发明,有可能自动地确定难以如列白名单所要求的那样定义的并且依赖于通过用户的目视观察的部分。

[0100] 更进一步地,根据本发明,因为通过不仅获取DOM树还获取CSS而比较图形用户接口(GUI),所以检测关于设计(例如行偏移量、画图失败、或重叠的项目)的问题也是可能的。具体地,根据本发明,因为通过不仅获取DOM树还获取CSS而比较GUI,所以在不可见的区域中的差异或者归因于显示大小的差异可以被检测。

[0101] 根据本发明,在对DOM树和CSS中的将被比较的内容的定义中,以及在用于DOM树和CSS的比较操作中,用户的参与被限制到仅为用于启动仿真的用户输入操作(见下文描述的图2中所示的过程),以及因此,比较是准确的和可重现的。

附图说明

[0102] 图1是示出可以被用在本发明的实施例中的计算机的示例的简图；

[0103] 图2示出根据本发明实施例的过程的流程图，该过程保存应用上的用户的输入操作以及用于至少一个期待检查屏幕状态时的定时中的每一个定时的DOM树和CSS，从而检测由改变从中可以提取DOM树和CSS的应用的源代码而引起的影响；

[0104] 图3示出根据本发明实施例的过程的流程图，该过程在图2中所示的流程图的完成以后，在按照操作顺序仿真在图2中执行的输入操作的期间保存DOM树和CSS，以及检测由改变应用的源代码而引起的影响；

[0105] 图4A是根据本发明实施例的过程的流程图，该过程将已经被保存的DOM树中的每一个DOM元素和CSS特性的值，与已经被获取的DOM树中的每一个DOM元素和CSS特性的值相比较，以及在图3中所示的流程图的步骤中输出比较的结果；

[0106] 图4B是根据本发明实施例的过程的流程图，该过程将已经被保存的DOM树中的每一个DOM元素和CSS特性的值，与已经被获取的DOM树中的每一个DOM元素与CSS特性的值相比较，以及在图3中所示的流程图的步骤中输出比较的结果；

[0107] 图5A示出根据本发明实施例的被保存的输入操作历史的存储模式；

[0108] 图5B示出根据本发明实施例的要被保存的DOM树的存储模式，以及要被应用到DOM树中的每一个DOM元素上的CSS（包含CSS特性和CSS特性的定义位置）；

[0109] 图6A示出根据本发明实施例的对由上文提到的改变而引起的影响的检测，以及该检测结果的示例输出；

[0110] 图6B示出根据本发明实施例的对由上文提到的改变而引起的影响的检测，以及该检测结果的示例输出；

[0111] 图6C示出根据本发明实施例的对由上文提到的改变而引起的影响的检测，以及该检测结果的示例输出；

[0112] 图7A示出根据本发明实施例的对由上文提到的改变而引起的影响的检测，以及该检测结果的示例输出；

[0113] 图7B示出根据本发明实施例的对由上文提到的改变而引起的影响的检测，以及该检测结果的示例输出；

[0114] 图7C示出根据本发明实施例的对由上文提到的改变而引起的影响的检测，以及该检测结果的示例输出；

[0115] 图8A示出根据本发明实施例的对由上文提到的改变而引起的影响的检测，以及该检测结果的示例输出；

[0116] 图8B示出根据本发明实施例的对由上文提到的改变而引起的影响的检测，以及该检测结果的示例输出；

[0117] 图8C示出根据本发明实施例的对由上文提到的改变而引起的影响的检测，以及该检测结果的示例输出；

[0118] 图9是示出根据本发明实施例，用于检测由上文提到的改变而引起的影响的计算机的功能模块简图的示例的简图，该计算机优选地具有图1的硬件配置；以及

[0119] 图10是示出根据本发明实施例，用于检测由上文提到的改变而引起的影响的计算

机的功能模块简图的示例的简图,该计算机优选地具有图1的硬件配置。

具体实施方式

[0120] 下文将参考附图描述本发明的实施例。下文描述的附图中,相同的参考符号被应用到相同的对象上,除非另有说明。本发明的实施例是用于描述本发明的优选的模式,并且应当被理解为不将本发明的范围限制为在此所描述的那样。

[0121] 可以被使用在本发明的实施例中的计算机不被具体地限制,只要该计算机能够检测由改变应用的源代码而引起的影响,其中DOM树和CSS可以从该应用提取。例如,此计算机可以是台式计算机、笔记本计算机、一体化计算机、平板或智能电话终端(例如加载有**Windows**[®]、Android或iOS的平板或智能电话)。

[0122] 在本发明的实施例中,“DOM树”表示树结构的XML或HTML文档。例如,XML文档逻辑上具有树结构并且包含根(首先出现的根元素)和从根延续的一个或多个节点元素或属性(分支或叶)。使用DOM,所读的XML文档被转换到树结构。树结构的数据形成DOM树。例如,一些网络应用(例如网络浏览器)包含访问被显示的页面的DOM树以及显示或编辑该DOM树的功能,作为标准的或附加组件。这些是例如Google**Chrome**[®] Developer Tools、或Internet Explorer(注册商标)的开发者工具、或用于**Firefox**[®]的附加组件**Firebug**[®]。

[0123] 在本发明的实施例中,“CSS”是由W3C发布的规范中的一个,其用于指示XML或HTML文档的元素如何被修饰的(或显示的),并且是关于例如字体、字体大小、字体修饰、行间距等的视觉效果的信息。例如,一些网络应用(例如网络浏览器)包含访问被显示的页面的CSS以及显示或编辑该CSS的功能,作为标准的或附加组件。这些是例如Google**Chrome**[®] Developer Tools、或Internet Explorer(注册商标)的开发者工具、或用于**Firefox**[®]的附加组件**Firebug**[®]。在此规范中,“CSS”可以是父元素继承的一个。

[0124] CSS可以包含CSS特性(CSS特性还是设置项目)、和CSS特性的值。例如,在定义“font-size:12px;”中,该CSS特性是“font-size”以及对该CSS特性定义的值是“12px”。

[0125] 在本发明的实施例中,“可以从中提取DOM树和CSS的应用”可以是任何应用,只要DOM树和CSS可以从应用中提取。换句话说,“可以从中提取DOM树和CSS的应用”是通过使用DOM树和CSS而被配置的任意的应用。

[0126] 具体地,“可以从中提取DOM树和CSS的应用”可以是任意的动态的应用,该应用是通过框架(例如**JavaScript**[®]框架)、组件(例如由应用开发者提供的组件)或其结合被动地构建。**JavaScript**[®]框架可以是Dojo Toolkit、jQuery或prototype.js。上文提到的任意的应用可以是在网络上(例如互联网或内部网)使用的网络应用。

[0127] 在许多情况中,网络应用通过在操作在网络浏览器上的基于编程语言(例如上文提到的**JavaScript**[®])的程序和网络服务器侧的程序之间的合作而操作。例如用户可以在网络浏览器上运行网络应用。网络浏览器可以是Internet**Explorer**[®](IE)、Google

Chrome[®]、Safari[®]、Firefox[®]等。

[0128] 网络应用可以作为用于博客(“blog”)、网络布告栏、网上银行、网上交易、网上购物等的应用而操作。

[0129] 在本发明的实施例中,“可以从中提取DOM树和CSS的应用的源代码”是构建该可以从中提取DOM树和CSS的应用的框架(例如**JavaScript[®]**框架)和组件(例如由应用开发者提供的组件)。

[0130] 在应用的源代码被改变的情况中,如果在不同的时间得到相同的DOM树和CSS,那么被显示在显示屏幕上的结果、在可不见区域里的差异和在显示大小方面的差异至少在相同的网络应用(和相同的版本)中将是相同的。

[0131] 图1是示出可以被用在本发明的实施例中的计算机的示例的简图。

[0132] 计算机(101)包含CPU(102)和主存储器(103),并且这些被连接到总线(104)。CPU(102)优选地基于32比特或64比特架构。例如,CPU(102)可以是英特尔公司的Core[™] i系列、Core[™] 2系列、Atom[™]系列、**Xeon[®]**系列、**Pentium[®]**系列或**Celeron[®]**系列、或者是超微半导体公司(AMD)的A系列、Phenom[™]系列、**Athlon[®]**系列、**Turion[®]**系列、或**Sempron[®]**、或者是国际商业机器公司的Power[™]系列。

[0133] 例如液晶显示器(LCD)的显示器(106)可以经由显示控制器(105)被连接到总线(104)。此外,液晶显示器(LCD)可以是例如触摸面板显示器或浮动触控显示器。显示器(106)可以被用来在恰当的图形接口上显示对象,该对象可以通过运行在计算机(101)上的软件(例如根据本发明的实施例的计算机程序或运行在计算机(101)上的多个计算机程序)的操作而被显示。此外,显示器(106)可以输出根据本发明的实施例的DOM树、CSS和比较的结果。

[0134] 例如硬盘或固态硬盘(SSD)的盘(108)可以可选地经由例如SATA或IDE控制器(107)被连接到总线(104)。

[0135] 例如CD、DVD或BD驱动器的驱动器(109)可以可选地经由例如SATA或IDE控制器(107)被连接到总线(104)。

[0136] 键盘(111)和鼠标(112)可以可选地经由例如外围装置控制器(110)(例如经由键盘鼠标控制器或USB总线)被连接到总线(104)。

[0137] 例如**Windows[®]**、**Mac OS[®] X**、**Linux[®]**、Android或iOS的操作系统、例如J2EE、**Java[®]**应用、**Java[®]**虚拟机(VM)、**Java[®]** just-in-time(JIT)编译器的**Java[®]**处理环境、根据本发明的实施例的计算机程序和其他程序、以及数据以能够被加载到主存储器(103)上的方式被存储在盘(108)中。

[0138] 盘(108)可以被构建到计算机(101)中、可以以允许被计算机(101)访问的方式经由电缆被连接、或者可以以允许被计算机(101)访问的方式经由有线的或无线的网络被连接。

[0139] 驱动器(109)可以按需被使用,以将例如操作系统的程序、应用或根据本发明的实施例的计算机程序从CD-ROM、DVD-ROM或BD安装在盘(108)上。

[0140] 通信接口 (114) 符合例如 **Ethernet**[®] 协议。通信接口 (114) 经由通信控制器 (113) 被连接到总线 (104), 起到以有线的或无线的方式将计算机 (101) 连接到通信线路 (115) 的作用, 并且提供用于计算机 (101) 的操作系统的通信功能的 TCP/IP 通信协议的网络接口层。此外, 通信线路可以是基于无线 LAN 连接标准的无线 LAN 环境、例如 IEEE802.11a/b/g/n 的 Wi-Fi 无线 LAN 环境、或移动电话网络环境 (例如 3G 或 4G 环境)。

[0141] 参考下文描述的图 2、3、4A 和 4B, 将根据本发明的实施例描述用于检测由改变从中可以提取 DOM 树和 CSS 的应用的源代码而引起的影响的流程图。此外, 在下文描述的图 2、3、4A 和 4B 的描述中, “应用” 指从中可以提取 DOM 树和 CSS 的应用。

[0142] 在下文中将使用术语 “执行记录”、“记录区段” 和 “定时信息”, 并且如下定义这些术语。

[0143] “执行记录” 指测试用例的整个尝试, 并且一次测试尝试是被定义为执行记录。例如, 一次测试尝试将下文描述的图 2 中所示的步骤 203 至 216 或下文描述的图 3 中所示的步骤 303 至 314 作为一个单元。关于一次测试尝试, 例如第一次测试尝试被表达为执行记录 1, 第二次测试尝试被表达为执行记录 2, 以及第三次测试尝试被表达为执行记录 3。

[0144] “记录区段” 是通过将测试用例的整个尝试的执行记录分成一组 “用户执行某操作并且按下用于 DOM 树和 CSS 的保存按钮” 的块而得到的单元。“记录区段” 被表达为按照记录的顺序的记录区段 i (i 是 0 和 n 之间的任意整数)。记录区段通过从用户接收用于保存 DOM 树和 CSS 的触发而被定义。记录区段 i 表达从第 $i-1$ 次触发操作至第 i 次操作的区段。照此, 记录区段 0 是容纳在初始状态中的 DOM 树和 CSS 的对应执行记录的区段, 在该初始状态中没有用户的输入操作。于是, 第一次触发至第二次触发被表达为记录区段 1, 第二次触发至第三次触发被表达为记录区段 2, 第三次触发至第四次触发被表达为记录区段 3, 以及第 $n-1$ 次触发至第 n 次触发被表达为记录区段 n , 其中 n 是最后一次触发。此记录区段 n 是用于在对应的执行记录中的记录区段的最大值。例如, 在下文描述的图 2 中, “记录区段 0” 通过步骤 203 和 204 指示, “记录区段 1” 通过步骤 205 至 208 指示, “记录区段 2” 通过步骤 209 至 212 指示, 以及 “记录区段 n ” 通过步骤 213 至 216 指示。即, 关于 “记录区段 i ”, 0 可以被表达为 “从测试目标应用的开始 (在纠正以前) 至第一次按下保存按钮的时刻”。此外, 关于 “记录区段 i ”, 1 至 n 之间的任意整数 i 可以被表达为 “第 $i-1$ 次按下保存按钮至第 i 次按下保存按钮或按下结束按钮的时刻”。相似地, 在图 3 中, 与图 2 中被保存的记录区段相符, “记录区段 0” 通过步骤 303 和 304 指示, “记录区段 1” 通过步骤 306 至 308 指示, “记录区段 2” 通过步骤 309 至 311 指示, 以及 “记录区段 n ” 通过步骤 312 至 314 指示。

[0145] “定时信息” 指关于由用户执行的操作的、用于再生顺序或操作间隔的时间信息。照此, 当提到 “定时信息 i ” 时, 这意味着在记录区段 i 中的由用户执行的操作的相对时间信息。所以, 通过用于 “定时信息 i ” 的 1 至 n 定义 i 。

[0146] 图 2 示出根据本发明实施例的过程的流程图, 该过程保存在应用上的用户的输入操作以及用于至少一个期待检查屏幕状态时的定时中的每一个定时的 DOM 树和 CSS, 从而检测由改变从中可以提取 DOM 树和 CSS 的应用的源代码而引起的影响。

[0147] 例如在针对回归测试来检测由改变作为测试目标的应用的源代码而引起的影响的情况下, 图 2 中所示的流程图的每一个步骤在执行回归测试 (图 3 和图 4A 和 4B 中所示的过程) 以前仅被执行一次。

[0148] 在步骤201中,计算机开始如下过程:保存应用上的用户的输入操作,以及保存用于至少一个期待检查屏幕状态时的定时中的每一个定时的DOM树和CSS。关于该应用,假定还没有做出对应用的源代码的改变。

[0149] 在步骤202中,计算机根据例如对来自用户的应用开始指令的接收而开始应用。

[0150] 在步骤203中,计算机从用户处接收用于保存应用的DOM树和CSS的触发,DOM树和CSS是处于初始状态中。对应的执行记录被此触发分割,并且相继的触发之间的区段被定义为记录区段。触发可以是对用户按下键盘的特定键(例如特定的功能键)的检测,该特定键被设置为初始状态保存按钮。计算机根据触发的接收,获取在按下特定键时的DOM树以及被应用到DOM树中的每个DOM元素的CSS。DOM树和被应用到DOM树中的每个DOM元素的CSS是通过计算机分析从网络服务器来的响应而被自动地创建,并且与在按下特定键的时候被显示在浏览器上的内容对应(对下文描述的步骤206、210和214可以适用此描述),其中计算机通过使用显示屏幕上的浏览器以在浏览器中呈现每个元素来分析从网络服务器来的响应。已经被获取的DOM树和被应用到DOM树中的每个DOM元素的CSS的示例在下文描述的图5B中示出。

[0151] 在步骤204中,计算机将在步骤203中已经获取的在初始状态中的DOM树和CSS中的每一个保存到存储介质(291)中。要被保存的CSS可以包含,对于DOM树中的每一个DOM元素,要被应用到DOM元素上的CSS特性以及CSS特性的定义位置。顺带而言,在要被应用到DOM树中的DOM元素上的CSS特性是未定义的情况中,计算机可以获取要作为用于每个DOM元素的初始值而被使用的值,并且在存储介质(291)中保存该值。要作为初始值而被使用的值不受在CSS文件中定义的值(即,实际写在CSS文件中的值)的限制。此外,计算机可以捕捉应用上的屏幕快照并且在存储介质(291)中保存该屏幕快照。存储介质(291)可以是例如图1中所示的盘(108)。在下文中,被保存的DOM树和CSS将被称作记录区段0(初始状态, $i=0$)的DOM树和CSS。

[0152] 在步骤205中,计算机接收当初始状态保存按钮在步骤203中已经被按下之后用户执行的一系列输入操作(下文将其称作输入操作1)。每次输入操作被执行时(例如在与输入操作被执行时相同的时间处),计算机可以接收输入操作,并且在存储介质(292)中保存该输入操作。输入操作包含通过键盘的输入操作、通过指向设备的输入操作、通过触摸面板的输入操作以及通过音频输入的输入操作。指向设备可以是例如鼠标、指向杆(**TrackPoint**[®])、触摸板、轨迹球或游戏手柄。

[0153] 在步骤206中,计算机检测被用户执行的保存按钮的按下(例如特定的功能键),该保存按钮与这是期待检查屏幕状态时的定时的事实关联。计算机在保存按钮的按下时获取DOM树和被应用到DOM树中的每个DOM元素的CSS。计算机例如根据对保存按钮的按下的检测,定义在先前的触发记录区段0($i=0$)之后的区段为记录区段1($i=1$)。

[0154] 在输入操作1在步骤205中没有被保存的情况中,计算机可以在步骤207中在存储介质(292)中保存输入操作1。存储介质(292)可以与存储介质(291)相同或不同,并且可以是例如图1中所示的盘(108)。此外,计算机在存储介质(292)中将关于输入操作1的每次输入操作的执行的定时信息保存为定时信息1(例如关于事件信息(例如协调信息、对元素(关于该元素已经执行了输入操作)的参照)输入操作的执行顺序的时间信息,以及输入操作之间的间隔)。输入操作1和定时信息1可以一起被称作输入操作历史1。

[0155] 在步骤208中,计算机在存储介质(292)中将在步骤206中获取的DOM树以及CSS中的每一个保存为关于记录区段1($i=1$)的信息。被保存的CSS可以包含,对于DOM树中的每一个DOM元素,要被应用到DOM元素上的CSS特性以及CSS特性的定义位置。顺带而言,在要被应用到DOM树中的DOM元素上的CSS特性是未定义的情况下,计算机可以获取要作为用于每个DOM元素的初始值而被使用的值,并且在存储介质(292)中保存该值。此外,计算机可以捕捉应用上的屏幕快照并且在存储介质(292)中保存该屏幕快照。在下文中,已经被保存的DOM树和CSS将被称作记录区段1($i=1$)的DOM树和CSS。计算机可以同时执行步骤207和208的过程,或者按照步骤207和步骤208的顺序,或者按照步骤208和步骤207的顺序。

[0156] 在步骤209中,计算机接收当保存按钮在步骤206中已经被按下之后用户执行的一系列输入操作(下文将其称为输入操作2)。每次输入操作被执行时(例如在与输入操作被执行时相同的时间处),计算机可以接收输入操作,并且在存储介质(292)中保存该输入操作。

[0157] 在步骤210中,计算机检测被用户执行的保存按钮的按下,该保存按钮与这是期待检查屏幕状态时的定时的事实关联。计算机在按下保存按钮的时间处获取DOM树和被应用到DOM树中的每个DOM元素的CSS。计算机例如根据对保存按钮的按下的检测,定义记录区段1($i=1$)之后的区段为记录区段2($i=2$)。

[0158] 在输入操作2在步骤209中没有被保存的情况下,计算机可以在步骤211中在存储介质(292)中保存输入操作2。此外,计算机在存储介质(292)中将关于输入操作2的每次输入操作的执行的定时信息保存为定时信息2。

[0159] 在步骤212中,计算机在存储介质(292)中将在步骤210中获取的DOM树以及CSS中的每一个保存为关于记录区段2($i=2$)的信息。被保存的CSS可以包含,对DOM树中的每一个DOM元素,要被应用到DOM元素上的CSS特性以及CSS特性的定义位置。顺带而言,在要被应用到DOM树中的DOM元素上的CSS特性是未定义的情况下,计算机可以获取要作为用于每个DOM元素的初始值而被使用的值,并且在存储介质(292)中保存该值。此外,计算机可以捕捉应用上的屏幕快照并且在存储介质(292)中保存该屏幕快照。在下文中,已经被保存的DOM树和CSS将被称作记录区段2($i=2$)的DOM树和CSS。计算机可以同时执行步骤211和212的过程,或者按照步骤211和步骤212的顺序,或者按照步骤212和步骤211的顺序。

[0160] 在步骤213中,计算机接收当保存按钮已经被用户执行的输入操作 $n-1$ 按下之后用户执行的一系列输入操作(下文将其称为输入操作 n)。每次输入操作被执行时(例如在与输入操作被执行时相同的时间处),计算机可以接收输入操作,并且在存储介质(292)中保存该输入操作。

[0161] 在步骤214中,计算机检测被用户执行的结束按钮(例如特定的功能键)的按下,该结束按钮与期待检查屏幕状态时的定时的结束关联。计算机在按下结束按钮的时间处获取DOM树和被应用到DOM树中的每个DOM元素的CSS。

[0162] 在输入操作 n 在步骤213中没有被保存的情况下,计算机可以在步骤215中在存储介质(292)中保存输入操作 n 。此外,计算机在存储介质(292)中将关于输入操作 n 的每次输入操作的执行的定时信息保存为定时信息 n 。

[0163] 在步骤216中,计算机在存储介质(292)中将在步骤215中获取的DOM树以及CSS中的每一个保存为关于记录区段 n ($i=n$)的信息。被保存的CSS可以包含,对于DOM树中的每一个DOM元素,要被应用到DOM元素上的CSS特性以及CSS特性的定义位置。顺带而言,在要被应

用到DOM树中的DOM元素上的CSS特性是未定义的情况中,计算机可以获取要作为用于每个DOM元素的初始值而被使用的值,并且在存储介质(292)中保存该值。此外,计算机可以捕捉应用上的屏幕快照并且在存储介质(292)中保存该屏幕快照。在下文中,已经被保存的DOM树和CSS将被称作记录区段 $n(i=n)$ 的DOM树和CSS。计算机可以同时执行步骤215和216的过程,或者按照步骤215和步骤216的顺序,或者按照步骤216和步骤215的顺序。

[0164] 在步骤217中,计算机结束保存应用上的用户的输入操作以及保存用于至少一个期待检查屏幕状态时的定时中的每一个的DOM树和CSS的过程。

[0165] 在图2所示的示例中,对用户的输入操作被执行了 n 次的情况示出输入操作1、2、 $\dots$ 、 n 。但是,这没有排除用户的输入操作于输入操作1处或输入操作2处结束的情况,并且应当注意到最后的记录区段 $n(i=n)$ 是根据步骤214中所指示的结束按钮的按下而被确定的。

[0166] 图3示出根据本发明实施例的过程的流程图,该过程在图2中所示的流程图的流程完成以后,在按照操作顺序仿真在图2中执行的输入操作的期间保存DOM树和CSS,以及检测由改变应用的源代码而引起的影响。

[0167] 图3中所示的流程图的每个步骤是在源代码被改变以后、在针对用于构建应用的回归测试、单元测试或代码研究而执行对由改变应用的源代码而引起的影响的检测的情况中被执行的。

[0168] 并非源代码作者的测试者基于由源代码作者做出的对源代码的改变不影响情景的显示结果的假设,而执行回归测试,以便检查在改变以前的源代码和改变以后的源代码之间是否有差异。

[0169] 可以执行单元测试从而使得源代码作者可以容易地规定例如被由源代码作者做出的源代码的改变(内部改变)影响的网络应用的部分(例如外部输出)。

[0170] 像上文描述的单元测试一样,可以由缺乏经验的源代码作者在源代码开发的初级阶段执行代码研究,从而检查源代码的描述和应用的输出结果之间的关联。

[0171] 作为图3中所示的每个步骤的执行主体的计算机可以是与图2中所示的每个步骤的执行主体的计算机相同的计算机(见下文所述的图9),或者它可以是不同的计算机(见下文所述的图10)。

[0172] 在步骤301中,计算机开始如下过程:在按照操作顺序仿真在图2中执行的输入操作1、2、 $\dots$ 、 n 的期间保存DOM树和CSS,以及检测由改变应用的源代码而引起的影响。

[0173] 假定通过对在图2中被执行的应用的源代码做出改变而已经得到了应用。

[0174] 这个过程可以在例如已经对应用的源代码做出改变以后运行回归测试时开始。

[0175] 在步骤302中,例如,计算机根据对来自用户的应用开始指令的接收而开始应用。

[0176] 在步骤303中,计算机从用户接收用于保存DOM树和CSS的触发,该DOM树和CSS处在对应用的测试开始的初始状态中。该触发可以是对用户按下键盘的特定键(例如特定的功能键)的检测,该特定键被设置为初始状态保存按钮。计算机根据触发的接收,获取在按下特定键时的DOM树以及被应用到DOM树中的每个DOM元素的CSS(与图2中步骤203中的初始状态对应)。

[0177] 在步骤304中,计算机将在步骤303中已经获取的在初始状态中的DOM树的每一个和CSS保存到存储介质(392)中。针对DOM树的每一个DOM元素,要被保存的CSS可以包含要被

应用到DOM元素上的CSS特性和CSS特性的定义位置。顺带而言,在要被应用到DOM树中的DOM元素上的CSS特性是未定义的情况下,计算机可以获取要作为用于每个DOM元素的初始值而被使用的值,并且在存储介质(392)中保存该值。此外,计算机可以捕捉应用上的屏幕快照并且在存储介质(392)中保存该屏幕快照。存储介质(392)可以是例如图1中所示的盘(108)。在下文中,被保存的DOM树和CSS将被称作记录区段0(初始状态, $i=0$)的DOM树和CSS。

[0178] 在步骤305中,计算机从存储介质(292)中提取输入操作历史(输入操作历史1至n)。

[0179] 在步骤306中,计算机从在步骤305中被提取的输入操作历史1中提定时信息1。

[0180] 在步骤307中,计算机从在步骤305中被提取的输入操作历史1中提取输入操作1,以及根据在步骤306中被提取的定时信息1仿真并且执行输入操作1。计算机根据输入操作1的仿真的结束来获取在结束时间(即定时信息1)处的DOM树以及将被应用到树中的每一个DOM元素上的CSS。

[0181] 在步骤308中,计算机将定时信息1的DOM树和CSS中的每一个保存到存储介质(392)中。要被保存的CSS可以包含,对于DOM树中的每一个DOM元素,要被应用到DOM元素上的CSS特性以及CSS特性的定义位置。顺带而言,在要被应用到DOM树中的DOM元素上的CSS特性是未定义的情况下,计算机可以获取要作为用于每个DOM元素的初始值而被使用的值,并且在存储介质(392)中保存该值。此外,计算机可以捕捉应用上的屏幕快照并且在存储介质(392)中保存该屏幕快照。在下文中,已经被保存的DOM树和CSS将被称作记录区段1($i=1$)的DOM树和CSS。

[0182] 在步骤309中,计算机从在步骤305中被提取的输入操作历史2中提取定时信息2。

[0183] 在步骤310中,计算机从在步骤305中被提取的输入操作历史2中提取输入操作2,以及根据在步骤309中被提取的定时信息2仿真并且执行输入操作2。计算机根据输入操作2的仿真的结束来获取在结束时间(即定时信息2)处的DOM树以及将被应用到树中的每一个DOM元素上的CSS。

[0184] 在步骤311中,计算机将在步骤310中已经被获取的记录区段2($i=2$)中的DOM树和CSS的每一个保存到存储介质(392)中。要被保存的CSS可以包含,对于DOM树中的每一个DOM元素,要被应用到DOM元素上的CSS特性以及CSS特性的定义位置。顺带而言,在要被应用到DOM树中的DOM元素上的CSS特性是未定义的情况下,计算机可以获取要作为用于每个DOM元素的初始值而被使用的值,并且在存储介质(392)中保存该值。此外,计算机可以捕捉应用上的屏幕快照并且在存储介质(392)中保存该屏幕快照。在下文中,已经被保存的DOM树和CSS将被称作记录区段2($i=2$)的DOM树和CSS。

[0185] 在步骤312中,计算机从在步骤305中被提取的输入操作历史n中提取定时信息n。

[0186] 在步骤313中,计算机从在步骤305中被提取的输入操作历史n中提取输入操作n,以及根据在步骤312中被提取的定时信息n仿真并且执行输入操作n。计算机根据输入操作n的仿真的结束来获取在结束时间(即定时信息n)处的DOM树以及将被应用到树中的每一个DOM元素上的CSS。

[0187] 在步骤314中,计算机将在步骤313中已经被获取的记录区段n($i=n$)中的DOM树和CSS的每一个保存到存储介质(392)中。要被保存的CSS可以包含,对于DOM树中的每一个DOM

元素,要被应用到DOM元素上的CSS特性以及CSS特性的定义位置。顺带而言,在要被应用到DOM树中的DOM元素上的CSS特性是未定义的情况下,计算机可以获取要作为用于每个DOM元素的初始值而被使用的值,并且在存储介质(392)中保存该值。此外,计算机可以捕捉应用上的屏幕快照并且在存储介质(392)中保存该屏幕快照。在下文中,已经被保存的DOM树和CSS将被称作记录区段 $n(i=n)$ 的DOM树和CSS。

[0188] 在步骤315中,关于相同的记录区段($i=i'$),计算机将在图2中所示的步骤204、208、212和216的每一个中已经保存的DOM树和CSS(在图3的步骤315中所示的“保存的DOM树和CSS”)与在图3中所示的步骤304、308、311和314的每一个中已经被保存的DOM树和CSS(在图3的步骤315中所示的“获取的DOM树和CSS”)比较。将参考下文描述的图4A和4B描述比较内容的细节和比较结果的输出。

[0189] 在步骤315中,计算机可以具体地将DOM树中的每个DOM元素与对应的DOM树中的每个对应的DOM元素比较。此外,具体地,计算机可以将CSS中的每一个CSS特性的每一个值与对应CSS中的每一个对应CSS特性的每一个值比较。进一步地,例如计算机还可以取已经从父元素继承的CSS作为比较目标。

[0190] 此外,在图3中所示的流程图中,在步骤313完成以后计算机共同地执行步骤315中的比较步骤,在步骤313处被仿真的输入操作是最后的输入操作。代替在步骤313之后共同地执行步骤315中的比较步骤,计算机可以在步骤304之后、步骤308之后、步骤311之后以及步骤314之后,即每次DOM树和CSS被保存之后执行比较步骤。或者,计算机可以根据用于输出比较结果的来自用户的对输出命令的输入,来执行步骤315中的比较步骤。

[0191] 在步骤316中,计算机结束此过程:在按照操作顺序仿真在图2中执行的输入操作的期间保存DOM树和CSS,以及检测由改变应用的源代码而引起的影响。

[0192] 用户可以在每次源代码被改变时重复地执行图3中所示的每一个过程。

[0193] 图4A和4B示出根据本发明实施例的过程的流程图,该过程在图3中所示的步骤315中执行,该过程将已经被保存的DOM树中的每一个DOM元素和CSS特性的值,与已经被获取的DOM树中的每一个DOM元素和CSS特性的值相比较,以及输出比较的结果。

[0194] 作为图4A和4B中所示的每一步骤的执行主体的计算机与图3中所示的每一步骤的执行主体的计算机相同。

[0195] 在步骤401中,计算机开始比较并且输出比较结果的过程。

[0196] 在步骤402中,计算机从存储介质(291,292)中读取用于由用户指示的记录区段 i 的DOM树和CSS,其中在存储介质(291,292)中已经保存了DOM树和CSS。此外,计算机从存储介质(392)中读取用于记录区段 i' 的DOM树和CSS,其中记录区段 i' 是与记录区段 i 相同的记录区段(即 $i=i'$),在存储介质(392)中已经保存了获取的DOM树和CSS。

[0197] 在步骤403中,计算机开始针对DOM树中的每一个DOM元素重复地执行由步骤404至409所指示的过程,其中该每一个DOM元素是确定目标。

[0198] 在步骤404中,计算机确定:关于DOM树中的每一个DOM元素,在相同的记录区段($i=i'$)里的DOM元素是否彼此相符,其中该每一个DOM元素是确定目标。如果DOM元素彼此相符,计算机继续进行至步骤405。在另一方面,如果元素的结合不相符,计算机继续进行至步骤408。

[0199] 此外,在步骤404中,在用于忽视DOM树中特定的DOM元素的比较的排除条件被设置

的情况中,计算机可以忽视DOM元素的比较。排除条件可以是不比较存在于DOM树中特定的标记下的DOM元素,或者不比较不被当作应用上的图形用户接口上的不同部分而注意的部分。例如,用户可以设置用于这样的部分的排除条件,关于该部分由功能的增加或源代码中的改变而引起的影响可以提前被预测(例如DOM元素、特定的位置或行号)。例如,用户可以设置排除条件从而使得在这样的DOM元素下的任何DOM元素被忽略,其中所述这样的DOM元素对源代码的增加是已知的(例如<div id="newFeature">)。计算机可以设置有助于接收排除条件的输入的用户接口(例如窗口菜单)。

[0200] 在步骤405中,计算机开始针对作为确定目标的每一个CSS特性重复地执行由步骤406和407指示的过程。

[0201] 在步骤406中,计算机确定作为确定目标的CSS特性的值在相同的记录区段($i = i'$)中是否彼此相符。即计算机可以将将在CSS文件中没有定义记录区段 i 的每一个CSS特性的值或CSS特性的情况下在浏览器的默认中使用的值(被当作初始值使用的值),与在CSS文件中没有定义对应记录区段 i' 的每一个对应CSS特性的值或CSS特性的情况下在浏览器的默认中使用的值(被当作初始值使用的值)比较。特别地,计算机可以在记录区段 i 的CSS特性中规定在记录区段 i 中用于实际屏幕显示的第一CSS特性(例如字体大小),并且在记录区段 i' 的CSS特性中规定在记录区段 i' 中用于实际屏幕显示的、并且与第一CSS特性具有相同的类型的第二CSS特性(例如字体大小),并且执行比较,所述比较是关于第一CSS特性的值或者在CSS文件中没有定义第一CSS特性的值的情况下在浏览器的默认中使用的值(被当作初始值使用的值)与第二CSS特性的值或者在CSS文件中没有定义第二CSS特性的值的情况下在浏览器的默认中使用的值(被当作初始值使用的值)是否是相同的值的(例如12px)。此外,在两个CSS文件的对应CSS特性都没有被定义的情况中,计算机对于这两个情况都使用在浏览器的默认中使用的值。照此,在这两个情况中在默认中使用的值相符,并且计算机确定没有不同的部分。以此方式,响应于这些CSS特性的值之间的相符,计算机继续进行至步骤409。在另一方面,在这些CSS特性的值不相符的情况中,计算机继续进行至步骤407。

[0202] 在步骤407中,因为这些CSS特性的值不相符,所以计算机规定这些CSS特性的不同的部分(差异),即不相同的这些CSS特性的值。计算机可以标记不同的部分从而使得当在步骤411中输出时它们以突出的方式被显示。

[0203] 在步骤408中,因为DOM元素在相同的记录区段($i = i'$)中不相符,所以计算机规定在DOM树中的DOM元素中的不同的部分。计算机可以标记不同的部分从而使得当在步骤411中输出时它们以突出的方式被显示。

[0204] 在步骤409中,在由步骤406和407指示的过程没有对所有的CSS特性完成的情况中,计算机重复由步骤406和407指示的过程。在由步骤406和407指示的过程针对所有CSS特性都完成了的情况中,计算机继续进行至步骤410。

[0205] 在步骤410中,在由步骤404至409指示的过程没有针对所有DOM树中的所有DOM元素都完成的情况中,计算机重复由步骤404至409指示的过程。在由步骤404至409指示的过程针对所有DOM树中的所有DOM元素都完成了的情况中,计算机继续进行至步骤411。

[0206] 在步骤411中,计算机可以针对相同记录区段($i = i'$)的每一个显示在步骤407和408中规定的不同的部分。

[0207] 不同的部分以如下方式被显示,例如:计算机可以在显示相同的记录区段($i = i'$)

的DOM树的DOM元素的窗口中,用突出的方式标记在步骤408中已经规定的不同的部分(例如见图6A至6C所示的DOM树621和611,以及图7A至7C所示的DOM树721和761);和/或计算机可以在显示被应用于相同记录区段($i=i'$)的每一个DOM元素的CSS的屏幕显示中,用突出的方式标记在步骤407中已经被规定的不同的部分。例如图6C中所示的窗口显示这些CSS(631和671),以及图7A至7C中所示的窗口显示这些CSS(731和771)。

[0208] 在步骤411中,计算机可以显示应用在相同记录区段($i=i'$)的窗口(例如网络浏览器)中显示的内容。例如,计算机可以在用于相同的记录区段($i=i'$)的窗口中显示在图2中所示的步骤204、208、211和216中捕捉的屏幕快照和在图3中所示的步骤304、308、311和314中捕捉的屏幕快照。

[0209] 进一步地,在步骤411中,计算机可以规定用于相同记录区段($i=i'$)的应用的屏幕显示(例如窗口)的不同的部分。计算机可以说明已经作为相同记录区段($i=i'$)的不同的部分被规定的不同的部分,即,应用上的图形用户接口之间的不同的部分。

[0210] 在步骤411中,计算机可以显示在不可见区域中的差异,该差异没有被当做应用上的图形用户接口之间的不同的部分被示出。在不可见区域中的差异可以通过DOM树的DOM元素和/或CSS特性或CSS特性的值被规定。

[0211] 在不可见区域中的差异可以通过CSS特性或CSS特性的值被规定。不可见区域中的差异基于CSS特性的情况是例如CSS特性仅在CSS文件中的一个中存在并且没有用于浏览器的初始值的情况。不可见区域中的差异基于CSS特性的值的情况是CSS特性仅在CSS文件中的一个中存在并且存在用于浏览器的初始值并且这些值不相同的情况,或者相同的CSS特性在两个CSS文件中都存在并且这两个的值不相同的情况。

[0212] 在步骤411中,对相同的记录区段($i=i'$)的每一个,计算机可以显示CSS文件以及CSS文件的位置(例如行号),其中应用上的图形用户接口之间的不同的部分被定义在该位置处。具体地,计算机可以用突出的方式显示在记录区段*i*的CSS特性中的与第一CSS特性对应的CSS特性,其中第一CSS特性定义在一个特定定时时被用于实际的屏幕显示的值并且该CSS特性定义没有被用于实际的屏幕显示的值;以及以相同的方式,计算机可以用突出的方式显示在记录区段*i'*的CSS特性中的与第二CSS特性对应的CSS特性,其中第二CSS特性定义在与上文提到的一个特定定时相同的定时时被用于实际的屏幕显示的值并且该CSS特性定义没有被用于实际的屏幕显示的值。

[0213] 而且,在步骤411中,在图4A和4B中所示的过程用于单元测试而被执行的情况中,计算机可以输出用于单元测试的差异报告(例如通过屏幕输出、文件输出或打印输出)。

[0214] 在步骤412中,计算机确定比较结果的输出操作是否完成。在比较操作完成的情况中,计算机继续进行至步骤413并且结束过程。在另一方面,在继续进行比较操作的情况中,计算机返回操作至步骤402。例如,在不同的记录区段被用户选择的情况中比较操作被继续。

[0215] 在步骤413中,计算机结束比较以及输出比较结果的过程。

[0216] 图5A示出根据本发明的实施例被保存的输入操作历史的存储模式。

[0217] 屏幕(501)示出当期待检查应用的屏幕状态时,在某定时(记录区段*i*)时被保存的输入操作内容的存储模式。

[0218] DOM树(511)示出DOM元素(521、522和523)的层次结构。DOM元素(521)是根DOM元

素,并且DOM元素(522和523)是分支DOM元素。

[0219] 要执行关于图2中所示流程图的步骤207、211和215的、由图3中所示的步骤306和307、309和310以及312和313指示的仿真操作,必须记录关于与用户的输入操作有关的元素的所有信息。这些信息是例如识别号(id)、标签和/或样式、事件类型(例如“onClick”)、以及用于确定自从记录区段i的开始以来在多少秒以后哪个事件发生的相对时间和事件的顺序(例如,“在记录区段i中的相对时间:ss.sss秒”)。识别号和标签和/或样式用于记录这样的目标,其中关于该目标用户已经执行输入(例如已经被点击的按钮、或已经执行键输入的文本框)。样式可以在识别号或按钮的标签被输入数据改变的状况中被用来执行对“首先出现的宽度:100px;高度:40px的按钮”的识别。而且,在记录区段i中的输入操作历史是与DOM元素(523)关联的,该输入操作历史是例如被用户执行的各种事件(例如按钮的点击、按钮的双击、拖和放、悬停(例如鼠标悬停)、键输入)。照此,在执行由图3中所示的步骤306和307、309和310以及312和313指示的仿真操作的时候,通过使用图5A中所示的操作历史使得计算机能够规定某个按钮被点击的方式。在执行由图3中所示的步骤306和307、309和310以及312和313指示的仿真操作的时候,例如如果那些识别号相同,计算机可以仿真在该相同的识别号的元素上的点击,并且如果识别号被动态地改变,可以仿真在带有相同标签的按钮上的点击。但是,依赖于使用的JavaScript(注册商标),id属性可以被自动地创建。照此,使得能够选择要被用于规定的属性值(例如通过标签选择)。计算机可以显示用于允许用户执行选择的窗口。

[0220] 存储介质(541)存储在图2中所示的定时(步骤207、211和215)(542)时的每一个记录区段i中的输入操作历史0、1、2、...、n。

[0221] 图5B示出根据本发明的实施例被保存的DOM树的存储模式,以及将被应用到DOM树中的每一个DOM元素上的一个或多个CSS(包含CSS特性和CSS特性的定义位置)。

[0222] 屏幕(551)示出当期待检查应用的屏幕状态时,在某个定时(记录区段i)时的DOM树(561)和DOM树(561)中的DOM元素(571、572和573),以及被应用到DOM元素(571、572和573)上的CSS(581、582和583)。

[0223] DOM树(561)示出DOM元素(571、572和573)的层次结构。DOM元素(571)是根DOM元素,并且DOM元素(572和573)是分支DOM元素。

[0224] CSS(581)指示DOM元素(571)与CSS(581)关联。CSS(581)包含应用到DOM元素(571)上的CSS特性(581a)。CSS(581)的定义位置由CSS(581)中的“/Framework1/style1.css”(581b)指示。

[0225] CSS(582)指示DOM元素(572)与CSS(582)关联。CSS(582)包含应用到DOM元素(572)上的CSS特性(582a)。CSS(582)的定义位置由CSS(582)中的“/Framework1/style1.css”(582b)指示。

[0226] CSS(583)指示DOM元素(573)与CSS(583)关联。CSS(583)包含在不同的定义位置处的CSS(583-1)和CSS(583-2)。CSS(583-1)包含被应用到DOM元素(573)上的CSS特性(583-1a)。CSS(583-2)包含被应用到DOM元素(573)上的CSS特性(583-2a)。CSS特性(583-2a)在CSS(583-2)中被定义,但是因为框架的CSS(583-1)中的CSS特性(583-1a)比CSS特性(583-2a)更加优先,在CSS特性(583-2a)上划了删除线从而指示它没有被应用到DOM元素上。

[0227] 存储介质(591)存储一系列在图2中所示的定时(步骤203、206、210和214)时在记

录区段*i*的每一个中保存的DOM树,以及将应用到DOM树中的每一个DOM元素上的一个或多个CSS(步骤204、208、212和216)(592)。以相同的方式,在图3中,存储介质(591)存储一系列在图2中所示的定时时在记录区段*i*的每一个中保存的DOM树,以及将应用到DOM树中的每一个DOM元素上的一个或多个CSS(步骤304、308、311和314)(591)。

[0228] 图6A至6C、7A至7C以及8A至8C的每一个示出根据本发明的实施例的对由改变从中可以提取DOM树和CSS的应用(控制面板应用的一部分)的源代码而引起的影响的检测,以及检测结果的输出的示例。

[0229] 图6A至6C示出用于检测结果的输出屏幕(601),以及输出屏幕(601)在输出屏幕(601)的左侧示出:为图2中所示的过程的(初始)执行时间的(执行记录1:2013/10/0110:00:00)的记录区段5(*i*=5)的应用的预览屏幕显示窗口(611)、用于显示记录区段5(*i*=5)的DOM树的窗口(621)、以及用于显示应用到DOM树中的每一个DOM元素上的一个或多个CSS的窗口(631),以及在输出屏幕(601)的右侧示出:在除了图2中所示的过程的执行以外还执行了图3中所示的过程两次以后得到的第三次(执行记录3:2013/10/2016:30:00)的记录区段5(*i*'=5)的应用的预览屏幕显示窗口(651)、用于显示记录区段5(*i*'=5)的DOM树的窗口(661)、以及用于显示应用到DOM树中的每一个DOM元素上的一个或多个CSS的窗口(671)。

[0230] 输出屏幕(601)在输出屏幕(601)的左上侧示出:为图2中所示的过程的执行时间的初始(执行记录1:2013/10/0110:00:00)的应用的预览屏幕显示窗口(611)以及滚动条(691),其用于使得在预览屏幕显示窗口(611)中能够选择在除了图2中所示的过程的(初始)执行以外还执行了图3中所示的过程*x*次以后而得到的应用的第*x*+1次预览屏幕的显示。通过操作滚动条(691)(例如通过选择Prev按钮(返回按钮)或Next按钮(继续进行按钮)),用户可以在预览屏幕显示窗口(611)中显示用于比目前正在被显示的执行记录更旧的或更新的执行记录的预览屏幕。

[0231] 以相同的方式,输出屏幕(601)在输出屏幕(601)的右上侧示出:在除了图2中所示的过程的(初始)执行以外还执行了图3中所示的过程两次以后得到的第三次(执行记录3:2013/10/2016:30:00)的应用的预览屏幕显示窗口(651)以及滚动条(692),其用于使得在预览屏幕显示窗口(651)中能够选择在除了图2中所示的过程的(初始)执行以外还执行了图3中所示的过程*x*次以后而得到的应用的第*x*+1次预览屏幕的显示。通过操作滚动条(692)(例如通过选择Prev按钮(返回按钮)或Next按钮(继续进行按钮)),用户可以在预览屏幕显示窗口(651)中显示用于比目前正在被显示的执行记录更旧的或更新的执行记录的预览屏幕。

[0232] 在输出屏幕(601)上,在左上侧的预览屏幕显示窗口(611)可以作为比在右上侧的预览屏幕显示窗口(651)暂时在过去的应用的显示屏幕。

[0233] 进一步地,输出屏幕(601)包含滚动条(693),其使得能够选择关于哪个记录区段(记录区段*i*或记录区段*i*',*i*=*i*')的预览屏幕将被显示在左上侧的预览屏幕显示窗口(611)和右上侧的预览屏幕显示窗口(651)中。通过操作滚动条(693)(例如通过选择Prev按钮(返回按钮)或Next按钮(继续进行按钮)),用户可以在左上侧的预览屏幕显示窗口(611)或右上侧的预览屏幕显示窗口(651)中显示用于特定的记录区段(记录区段*i*(*i*=0至*n*)或记录区段*i*'(*i*'=0至*n*))的预览屏幕。

[0234] 关于输出屏幕(601)的描述以相同的方式被应用于图7A至7C所示的输出屏幕

(701)以及图8A至8C所示的输出屏幕(801)。

[0235] 在图6A中,预览屏幕显示窗口(611)示出用于初始(执行记录:2013/10/0110:00:00)的记录区段5($i=5$)的预览屏幕,该初始执行记录为执行记录1。预览屏幕显示窗口(651)示出用于第三次(执行记录:2013/10/2016:30:00)的记录区段5($i'=5$)的预览屏幕,该第三次执行记录是执行记录3。

[0236] 预览屏幕显示窗口(611)和预览屏幕显示窗口(651)之间的显示以如下方式而不同:

[0237] -“-----”的文本修饰(高度:1px(611a),高度2px(651a))

[0238] -“Hello,admin.”的显示位置(左对齐(611b),右对齐(651b));

[0239] -“Version”号(1.0(611c),1.02(651c))

[0240] -警告显示的有/无(“Warning:1 alert(s)” (651d)):“Warning:1 alert(s)”是加在控制面板应用上的头部分,与预览屏幕显示窗口(611)形成对照。此头部分“Warning:1 alert(s)”在图6A中所示的示例中是警告的显示,其指示关于控制面板应用中的状态C的警告被发出。

[0241] 在预览屏幕显示窗口(611)中,框部分(611c)是由改变DOM而引起的差异,以及框部分(611a,611b)是由改变CSS而引起的差异。在预览屏幕显示窗口(651)中,框部分(651c,651d)是由改变DOM而引起的差异,以及框部分(651a,651b)是由改变CSS而引起的差异。

[0242] 在预览屏幕显示窗口(611)和预览屏幕显示窗口(651)之间被不同地显示的部分(即,显示窗口的比较结果)以突出的方式被显示。被不同地显示的部分是被源代码中的改变影响的部分。在不同的部分是文本的情况下,突出的显示可以通过例如将该文本加框、修饰该文本(例如粗体、斜体、加下划线或改变颜色)、或通过加边界线分隔文本的行或列而被执行。在不同的部分是图像的情况下,突出的显示可以通过将图像加框、或通过加边界线分隔图像的行或列而被执行。

[0243] 在图6A中,显示DOM树的窗口(621)显示DOM元素树,其包含影响预览屏幕显示窗口(611)中的比较结果的DOM元素。例如,框部分(621a,621b,和621c)对应于影响预览屏幕显示窗口(611)中的不同的部分(611a,611b和611c)的DOM元素。框部分(621a,621b,和621c)可以通过被框部分环绕的部分的背景颜色不同而以突出的方式被显示。

[0244] 在图6A中,显示DOM树的窗口(661)显示DOM元素树,其包含影响预览屏幕显示窗口(651)中的比较结果的DOM元素。例如,框部分(661a,661b,661c和661d)对应于影响预览屏幕显示窗口(651)中的不同的部分(651a,651b,651c和651d)的DOM元素。

[0245] 在图6A中,用于显示CSS的窗口(631和671)尚未显示任何东西。

[0246] 例如,如图6B中所示,假定用户已经通过定点设备在用于显示DOM树的窗口(621)中的框部分(621a)上点击并且选择了该框部分,即<hr>标记,其为以突出的方式显示的DOM元素。如上文所描述的,框部分(621a)对应于影响在预览屏幕显示窗口(611)中显示的部分(611a)的DOM元素。根据在框部分(621a)上的点击,计算机以突出的方式在用于显示DOM树的窗口(621)中显示其为点击对象的DOM元素(621a)或包含DOM元素(621a)的外围的部分。例如,计算机改变<hr>标记(其为DOM元素(621a))的背景颜色,或者改变<hr>标记的字体。再者,计算机根据在框部分(621a)上的点击,在预览屏幕显示窗口(611)和预览屏幕显示窗口(651)的每一个中以突出的方式显示框部分(611a)和框部分(651a)。例如,计算机

改变“-----”的背景颜色或改变“-----”的字体。

[0247] 如图6C中所示,响应于DOM元素(621a)的选择,计算机检测在应用到DOM元素(621a)上的CSS和对应于DOM元素(621a)的DOM元素(661a)之间的差异,并且在用于显示CSS的窗口(631,671)中显示检测到的差异(631a,671a)。这些CSS之间的差异是在用于显示CSS的窗口(631)中的框(631a),以及在用于显示CSS的窗口(671)中的框(671a)。由窗口(631)中的框(631a)显示的“+height:1px;(default)”是通过在浏览器的默认中使用的值补充的,并且不基于在CSS文件中描述的内容。即,这不是简单地显示在CSS文件中描述的内容。照此,计算机比较以下两项:在用于显示CSS的窗口(631)中的“+height:1px;(default)”(其不基于在CSS文件中描述的内容)和在用于显示CSS的窗口(671)中的“+height:2px;”(其为在CSS文件中描述的内容),并且得到差异。计算机可以用突出的方式显示差异。突出显示的模式与上文所描述的模式相同。

[0248] 在图6C中所示的示例中,指示出了<hr>的高度被改变,这对于用户来说很难仅通过查看预览屏幕显示窗口(611)和预览屏幕显示窗口(651)而目测地检查。尽管关于检测到的差异(631a,671a)的CSS特性的名字是相同的,都是“高度”,但是由这些差异中的一个(631a)的CSS特性定义的值(实际上,描述的是(默认),并且该值是通过如下文描述的在浏览器的默认中使用的值而补充的值)是“1px”,并且由这些差异中的另一个(671a)的CSS特性定义的值是“2px”。所以,被定义的值在这些差异中的一个(631a)和这些差异中的另一个(671a)之间是不同的。

[0249] 在用于显示CSS的窗口(631)中,<hr>的CSS特性——高度——在CSS文件中未定义。照此,计算机通过使用1px而执行显示,其中1px是用于显示预览屏幕的应用(例如网络应用)的初始值。此初始值对每一个网络应用可以不同。照此,在用于显示CSS的窗口(631)中,描述了(default)从而补充用于与由CSS特性定义的值比较的初始值是什么。

[0250] 如图6C中所示,例如假定用户已经通过定点设备在CSS特性中的差异(671a)上点击并且选择了该差异(671a)。响应于对CSS特性中的差异(671a)的选择,计算机可以跳到文件被定义处的行。

[0251] 图7A中所示的输出屏幕(701)是响应于图6B中所示的DOM元素(621a)的选择而被显示的屏幕。照此,图7A中所示的输出屏幕(701)与图6C中所示的输出屏幕(601)相同。

[0252] 如图7B中所示,例如假定用户已经通过定点设备在用于显示DOM树的窗口(721)中的框部分(621b)(即以突出的方式显示的DOM元素(621b))上点击并且选择了该框部分(621b)。如上文描述,框部分(621b)对应于影响预览屏幕显示窗口(711)中显示的不同的部分(611b)的DOM元素。根据在框部分(621b)上的点击,计算机以突出的方式在用于显示DOM树的窗口(721)中显示其为点击对象的DOM元素(621b)或包含DOM元素(621b)的外围的部分。例如,计算机改变<pclass="oneuiSubTitle">Hello,admin.</p>的背景颜色,或者改变<p>标识的字体。再者,计算机根据在框部分(621b)上的点击,在预览屏幕显示窗口(711)和预览屏幕显示窗口(751)的每一个中以突出的方式显示框部分(611b)和框部分(651b)。例如,计算机改变的“Hello,admin.”(611b,651b)的背景颜色或改变“Hello,admin.”(611b,651b)的字体。此外,用于显示DOM树的窗口(721)中的框部分(621b)中的DOM元素与用于显示DOM树的窗口(761)中的框部分(661b)中的DOM元素之间没有差异。

[0253] 如图7C中所示,响应于DOM元素(621b)的选择,计算机检测在应用到DOM元素

(621a) 上的CSS和对应于DOM元素 (621b) 的DOM元素 (661b) 之间的差异,并且在用于显示CSS的窗口 (731,771) 中显示检测到的差异 (731b,771b)。这些CSS之间的差异是在用于显示CSS的窗口 (731) 中的框 (731b),以及在用于显示CSS的窗口 (771) 中的框 (771b)。计算机可以用突出的方式显示该差异。突出的显示的模式与上文所述的模式相同。

[0254] 在图7C中所示的用于显示CSS的窗口 (731) 中,关于将应用到DOM元素 (621b) 上的多个CSS (781a,781b),按带有更窄的范围和更高的优先级的元素安排这些CSS特性。而且,例如在对相同的DOM元素 (621b) 定义了相同的CSS特性的情况中,在并非将被应用的CSS特性的其它CSS特性上划了删除线(见781b)。

[0255] 在图7C中所示的用于显示CSS的窗口 (731) 一起显示CSS (781a) 的定义位置“/OneUI/OneUITheme.css”。而且,以相同的方式,用于显示CSS的窗口 (731) 一起显示多个CSS (781b) 的定义位置“/dojo/dijit/dijitTheme.css”。

[0256] 以相同的方式,用于显示CSS的窗口 (771) 中,关于应用到DOM元素 (661b) 上的多个CSS (791a,791b),按带有更窄的范围和更高的优先级的元素安排这些CSS特性。而且,例如在对相同的DOM元素 (661b) 定义了相同的CSS特性的情况中,在并非将被应用的CSS特性的其它CSS特性上划了删除线(见791b)。在用于显示CSS的窗口 (771) 中,显示了多个CSS (791a,791b) 的定义位置“/OneUI/OneUITheme.css”和“/dojo/dijit/dijitTheme.css”。

[0257] 在图7C中所示的示例中,应用到DOM元素<p>上的类“oneuiSubTitle”的CSS特性“text-align”在用于显示CSS的窗口 (731,771) 中被示为差异 (731b,771b)。关于检测到的差异 (731b,771b),由用于这些差异中的一个 (731b) 的CSS特性定义的值是“left”,以及由用于这些差异中的另一个 (771b) 的CSS特性定义的值是“right”。所以,对于这些差异中的一个 (731b) 和这些差异中的另一个 (771b),被定义的值是不同的。

[0258] 而且,应用到DOM元素<body>上的类“dijitText”的font-size在文件层级是不同的。即,对类“dijitText”其为“font-size:large;”(见781b),而对类“dijitText”其为“font-size:x-large;”(见791b)。但是,归因于关于CSS特性的优先级,代替类“dijitText”的font-size的值(见781b和791b),类“oneuiSubTitle”的font-size的值 (781a,791a) 被用于显示。照此,确定了类“dijitText”的font-size的值(见781b和791b)不影响在预览屏幕显示窗口 (711) 和预览屏幕显示窗口 (751) 上的显示的结果,并且这些值没有以突出的方式被显示。

[0259] 如图7C中所示,例如假定用户已经通过定点设备在CSS特性中的差异 (771b) 上点击并且选择了该差异 (771b)。响应于CSS特性中的差异 (771b) 的选择,计算机可以跳到文件被定义处的行。

[0260] 图8A至8C示出根据本发明的实施的由上文提到的改变而引起的影响的检测,以及检测结果的输出的示例。

[0261] 图8A至8C示出用于检测结果的输出屏幕 (801),以及输出屏幕 (801) 在输出屏幕 (801) 的左侧示出:为图2中所示的过程的(初始)执行时间的(执行记录1:2013/10/0110:00:00)的记录区段5 (i=5) 的应用的预览屏幕显示窗口 (811)、用于显示记录区段5 (i=5) 的DOM树的窗口 (821)、以及用于显示应用到DOM树中的每一个DOM元素上的一个或多个CSS的窗口 (831),以及在输出屏幕 (801) 的右侧示出:在除了图2中所示的过程的执行以外还执行了图3中所示的过程以后得到的第二次(执行记录2:2013/10/2014:20:00)的记录区段5

(i'=5)的应用的预览屏幕显示窗口(851)、用于显示记录区段5(i'=5)的DOM树的窗口(861)、以及用于显示应用到DOM树中的每一个DOM元素上的一个或多个CSS的窗口(871)。

[0262] 在图8A中示出的示例中,在预览屏幕显示窗口(811)上和预览屏幕显示窗口(851)上显示的内容之间没有差异。照此,没有用于指示在预览屏幕显示窗口(811)和预览屏幕显示窗口(851)中的差异的突出的显示。

[0263] 以相同的方式,在用于显示DOM树的窗口(821)中的DOM树中的每一个DOM元素和用于显示DOM树的窗口(861)中的DOM树中的每一个DOM元素之间没有差异。照此,没有用于指示用于显示DOM树的窗口(821)中和用于显示DOM树的窗口(861)中的差异的突出的显示。

[0264] 在图8B中示出的输出屏幕(801)与在图8A中示出的输出屏幕(801)相同。如在图8B中所示,假定用户已经通过定点设备在用于显示DOM树的窗口(821)中的DOM元素“<p class="oneuiSubTitle">Hello,admin.</p>”(框部分)上点击并且选择了该框部分。此框是加上去的从而清楚地指示被选择的部分,并且该框不必被显示在实际的屏幕上。

[0265] 如在图8C中所示,响应于DOM元素的选择,计算机检测在被应用到所选择的DOM元素上的CSS和对应于所选择的DOM元素的DOM元素“<p class="oneuiSubTitle">Hello,admin.</p>”(见861)之间的差异。计算机分别在用于显示CSS的窗口(831)和用于显示CSS的窗口(871)中显示被应用到所选择的DOM元素上的CSS以及被应用到与所选择的DOM元素对应的DOM元素上的CSS。但是,因为在CSS之间没有检测到差异,计算机在用于显示CSS的窗口(831)和用于显示CSS的窗口(871)中不用突出的方式显示CSS。

[0266] 图9是示出根据本发明实施例的用于检测由应用中的改变而引起的影响的计算机的功能模块简图的示例的简图,其中DOM树和CSS可以从该应用提取,该计算机优选地具有图1的硬件配置。

[0267] 根据本发明的实施例,计算机(901)是用于检测由改变应用的源代码而引起的影响的计算机,并且是例如图1中所示的计算机。

[0268] 计算机(901)包含保存部件(911)、获取部件(912)、比较部件(913)以及输出部件(914)。

[0269] 保存部件(911)在例如存储介质(921)(对应于图2中所示的存储介质(291))或存储介质(922)(对应于图2中所示的存储介质(292))中保存用户在应用上的输入操作以及用于当期待检查屏幕状态时的至少一个定时中的每一个定时的DOM树和CSS。

[0270] 保存部件(911)可以执行图2中所示的步骤203至216。

[0271] 源代码被改变以后,获取部件(912)按照操作的顺序仿真输入操作、获取用于与当期待检查屏幕状态时的至少一个定时中的每一个定时相同的定时的DOM树和CSS、并且在例如存储介质(923)(对应于图3中所示的存储介质(392))中保存该DOM树和CSS。

[0272] 获取部件(912)可以执行图3中所示的步骤303至314。

[0273] 比较部件(913)从例如存储介质(921)或存储介质(922)中读由保存部件(911)保存的DOM树和CSS,并且从例如存储介质(923)中读由获取部件(912)保存的DOM树和CSS。比较部件(913)对这些相同的定时的每一个,将从存储介质(921)或存储介质(922)中读取出的DOM树和CSS与从存储介质(923)中读取出的DOM树和CSS比较。具体地,比较部件(913)可以将DOM树中的每一个DOM元素与对应的DOM树中的每一个对应的DOM元素比较。进一步具体地,CSS中的每一个CSS特性的每一个值与对应CSS中的每一个对应CSS特性的每一个值可以

被互相比较。

[0274] 比较部件 (913) 可以执行图3中所示的步骤315,以及图4A中所示的步骤402至410和412。

[0275] 输出部件 (914) 输出比较部件 (913) 的比较结果。

[0276] 输出部件 (914) 可以执行图4B中所示的步骤411。

[0277] 图10是示出根据本发明实施例的用于检测由应用的改变而引起的影响的计算机的功能模块简图的示例的简图,其中DOM树和CSS可以从该应用提取,该计算机优选地具有图1的硬件配置。

[0278] 像图9中所示的计算机 (901) 一样,图10中所示的计算机 (1001) 包含获取部件 (1012)、比较部件 (1013) 以及输出部件 (1014)。提供给计算机 (1001) 的获取部件 (1012)、比较部件 (1013) 以及输出部件 (1014) 分别对应于提供给计算机 (901) 的获取部件 (912)、比较部件 (913) 以及输出部件 (914)。图10中所示的计算机 (1001) 与图9中所示的计算机 (901) 不同在于没有提供给计算机 (1001) 保存部件 (911)。

[0279] 计算机 (1001) 能够访问存储介质 (1021) (对应于图2中所示的存储介质 (291)) 和存储介质 (1022) (对应于图2中所示的存储介质 (292)),这两个存储介质保存用户在应用上的输入操作以及用于当期待检查屏幕状态时的至少一个定时中的每一个定时的DOM树和CSS。

[0280] 源代码被改变以后,获取部件 (1012) 访问存储介质 (1022)、按照操作的顺序仿真输入操作、获取用于与当期待检查屏幕状态时的至少一个定时中的每一个定时相同的定时的DOM树和CSS、并且在例如存储介质 (1023) (对应于图3中所示的存储介质 (392)) 中保存该DOM树和CSS。

[0281] 获取部件 (1012) 可以执行图3中所示的步骤303至314。

[0282] 比较部件 (1013) 对这些相同的定时的每一个,将保存在例如存储介质 (1021) 或存储介质 (1022) 中的DOM树和CSS与保存在例如存储介质 (1023) 中的DOM树和CSS比较。

[0283] 比较部件 (1013) 可以执行图3中所示的步骤315,以及图4A中所示的步骤402至410和412。

[0284] 输出部件 (1014) 输出比较部件 (1013) 的比较结果。

[0285] 输出部件 (1014) 可以执行图4B中所示的步骤411。

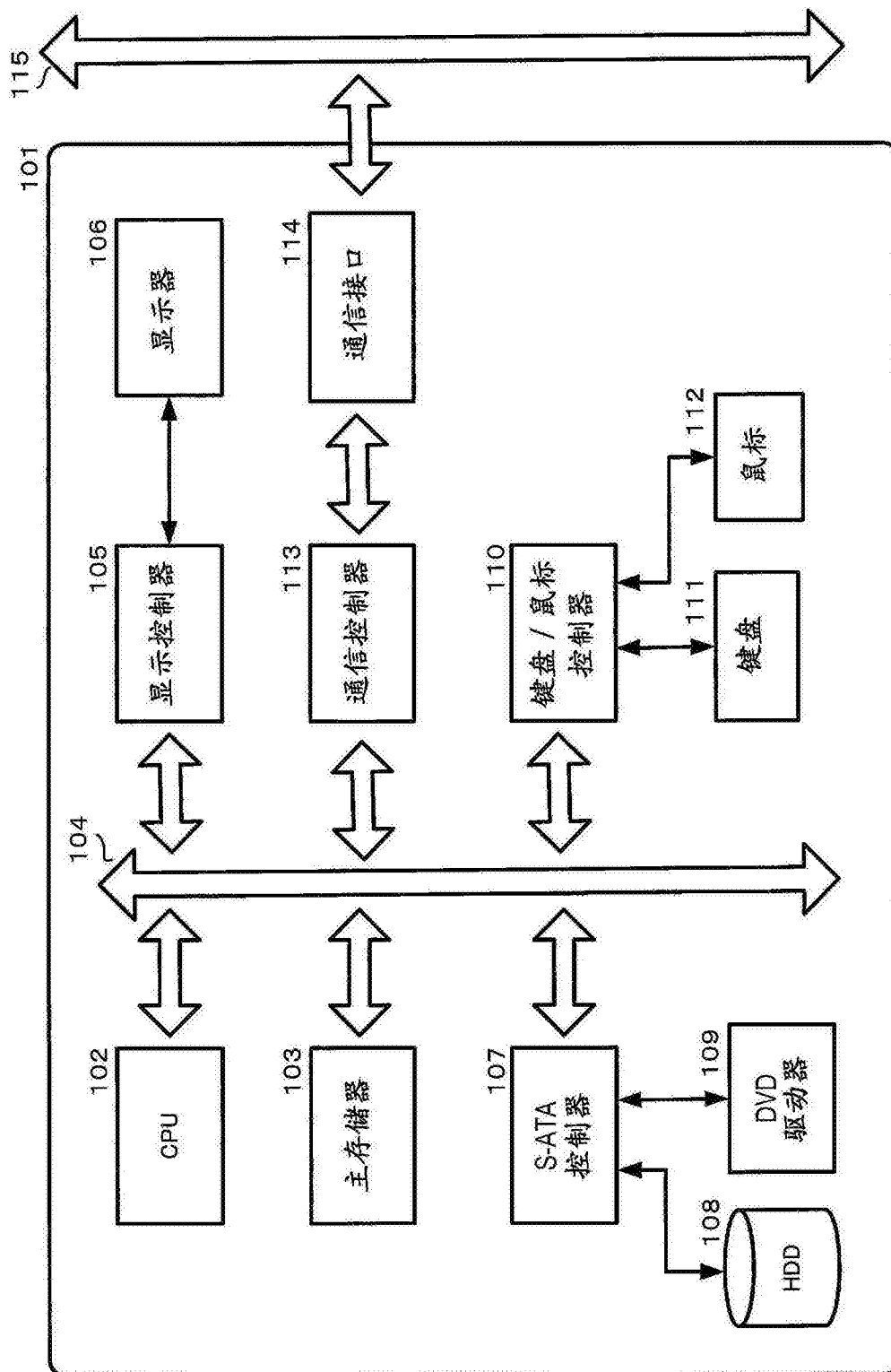


图1

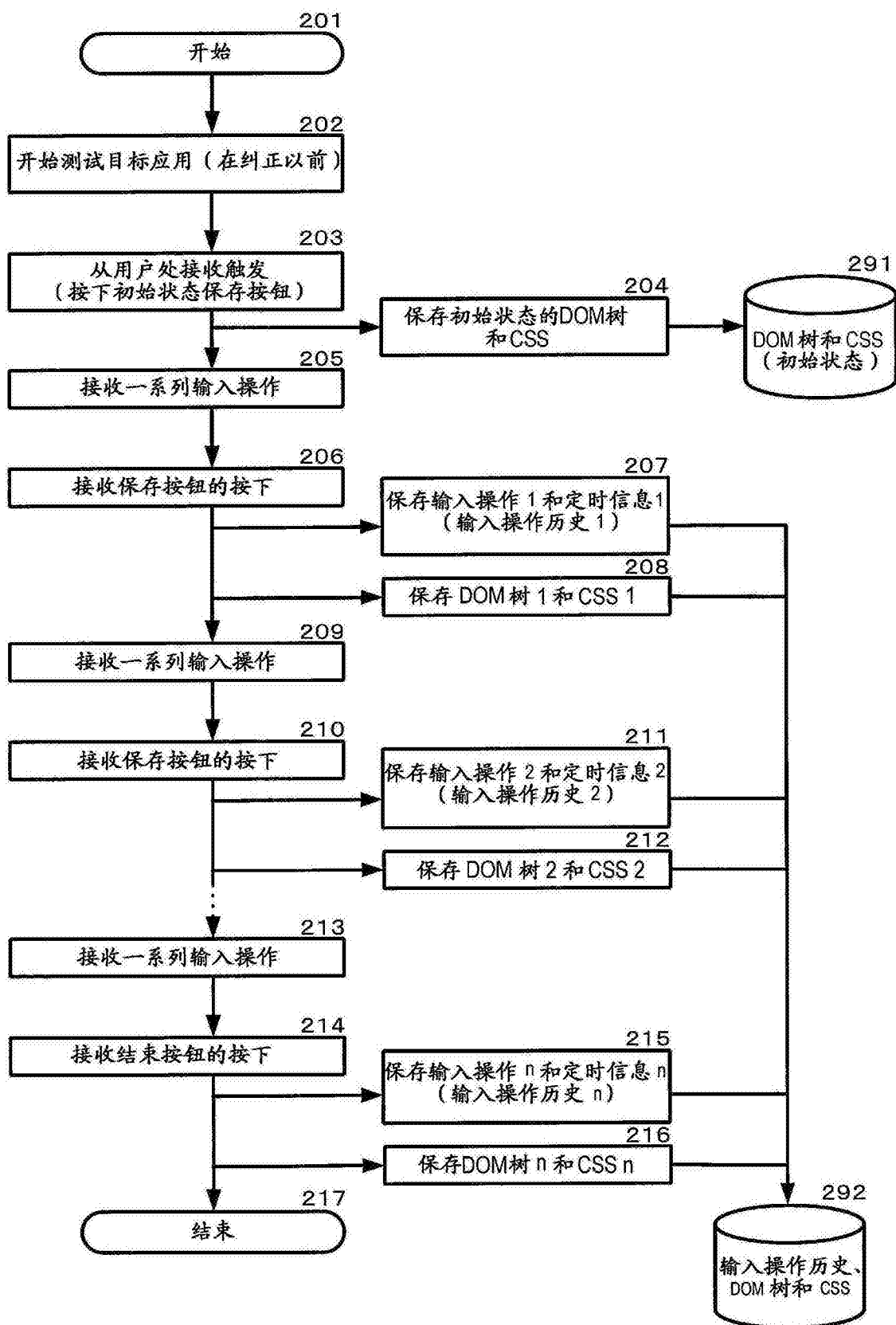


图2

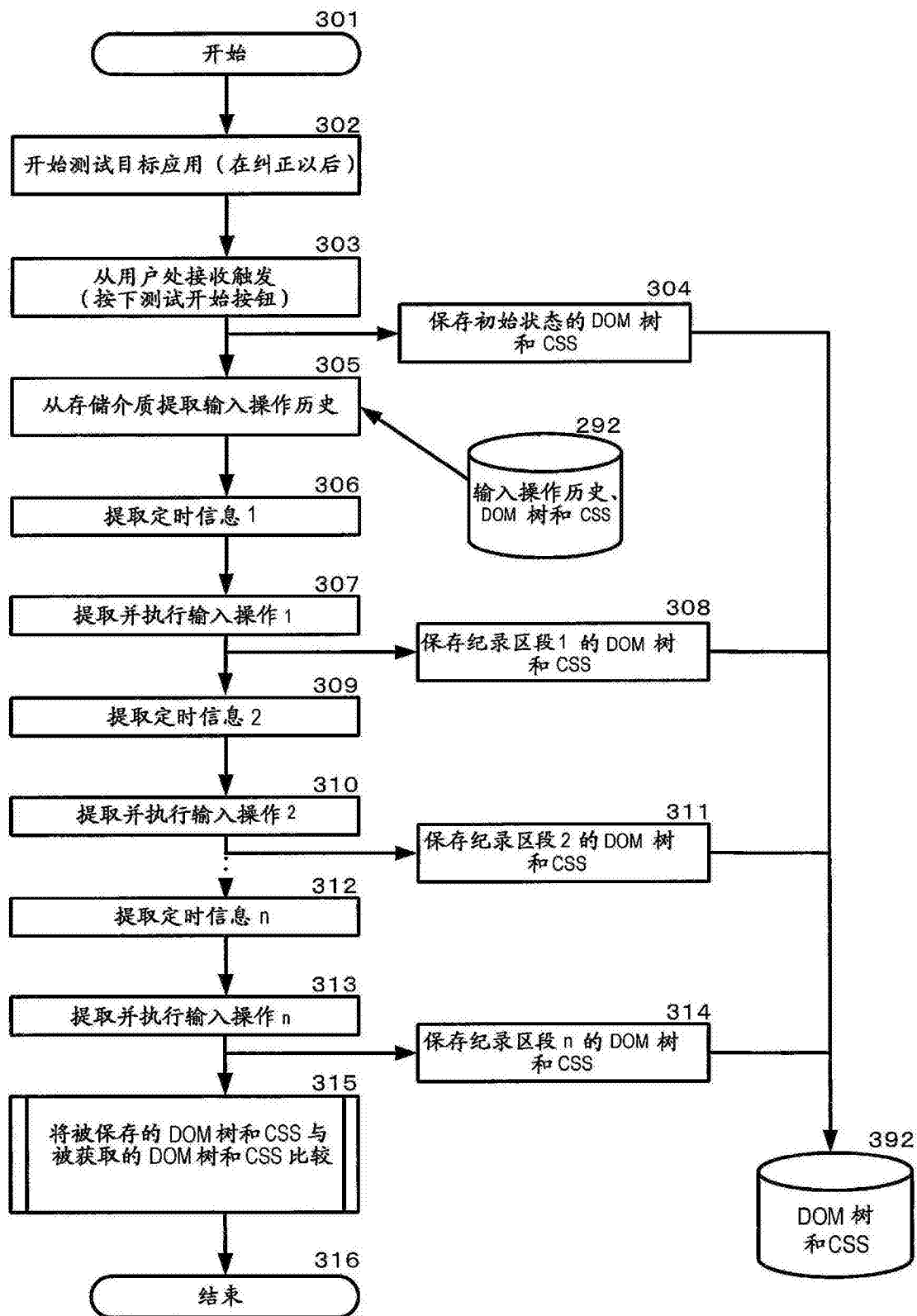


图3

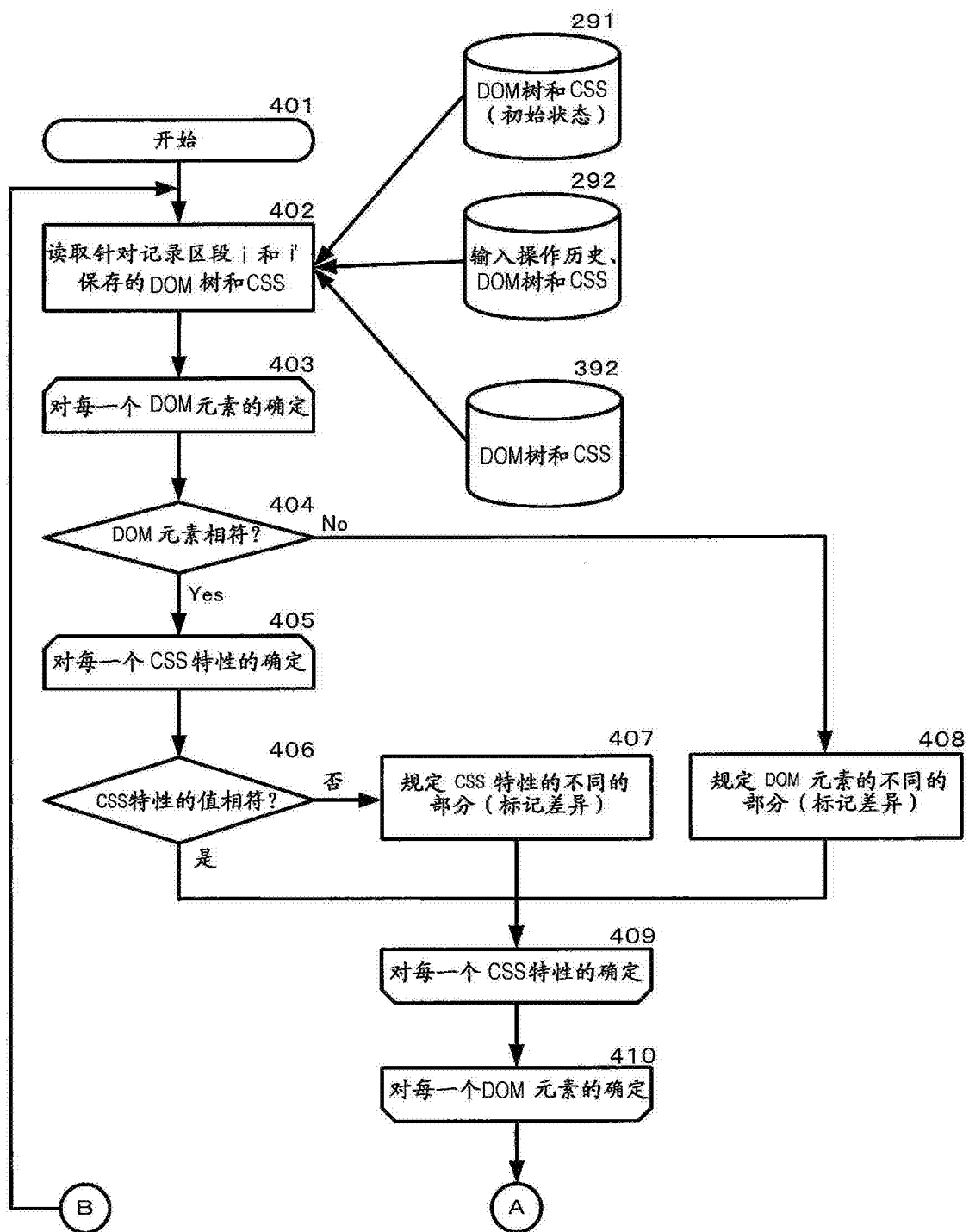


图4A

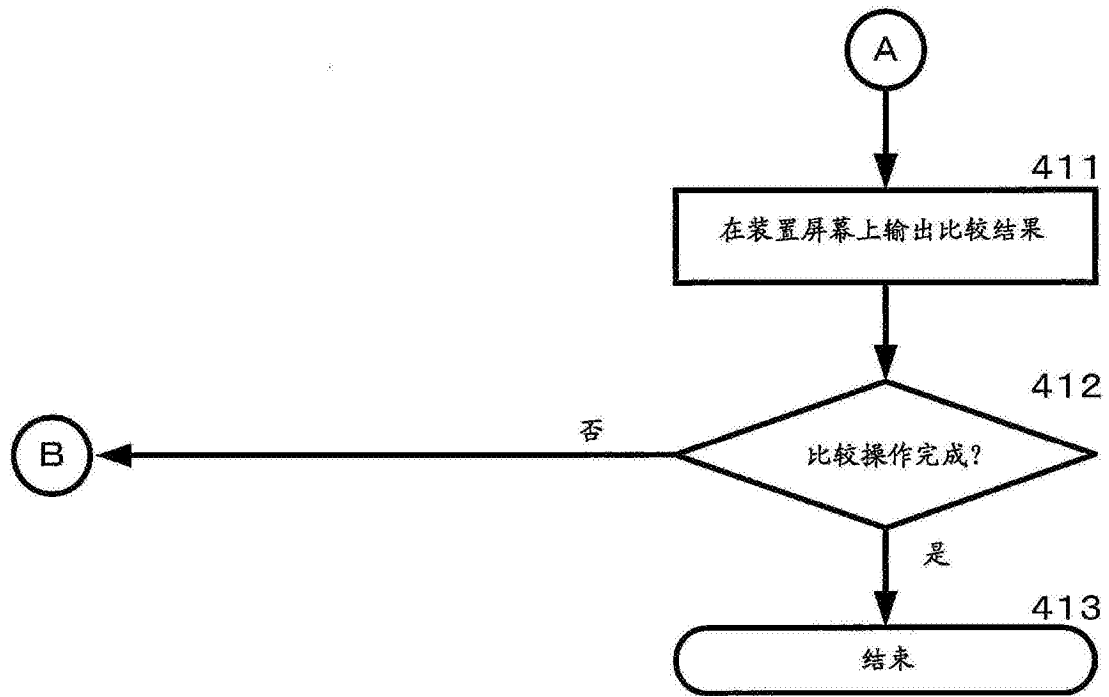


图4B

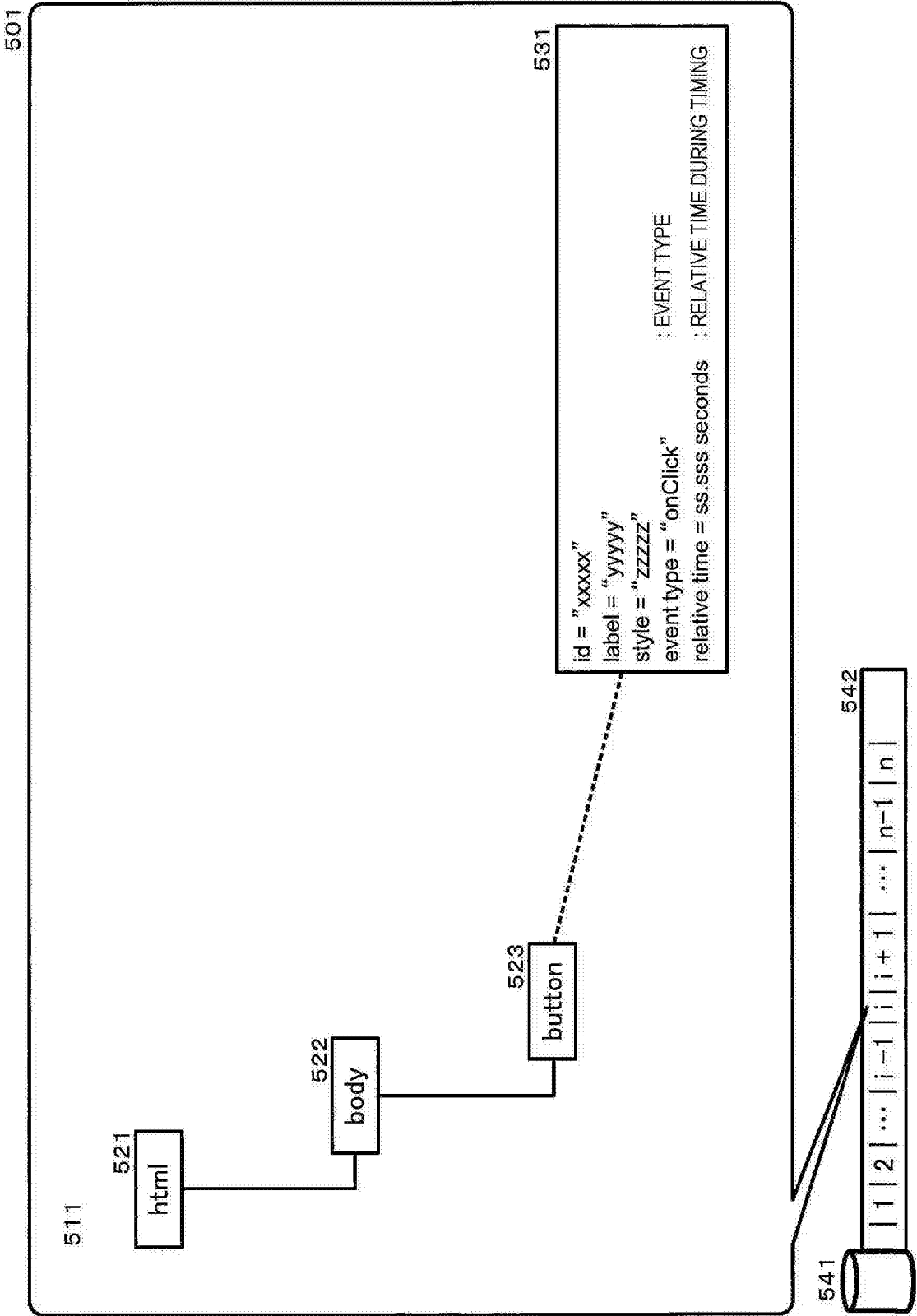


图5A

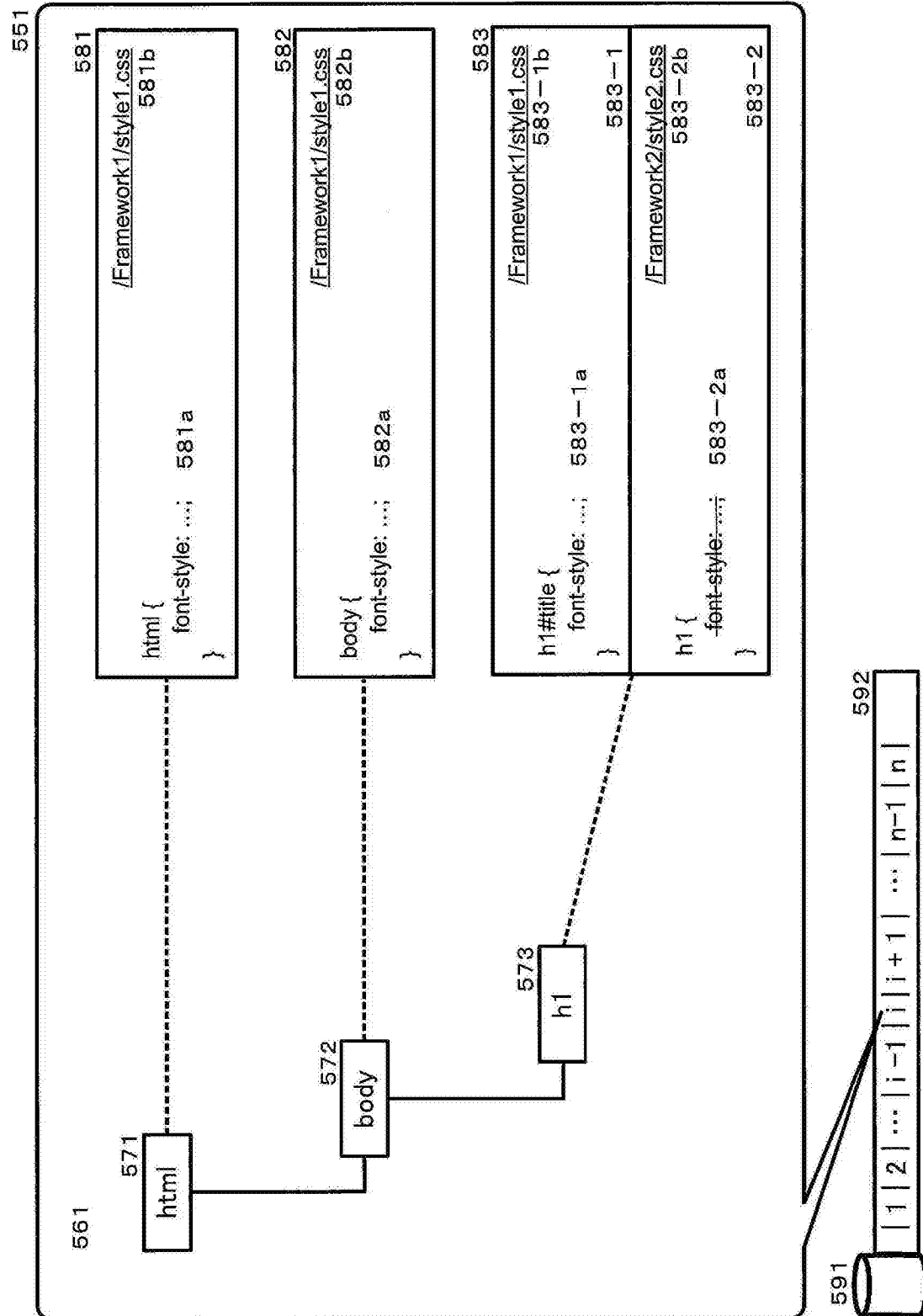


图5B

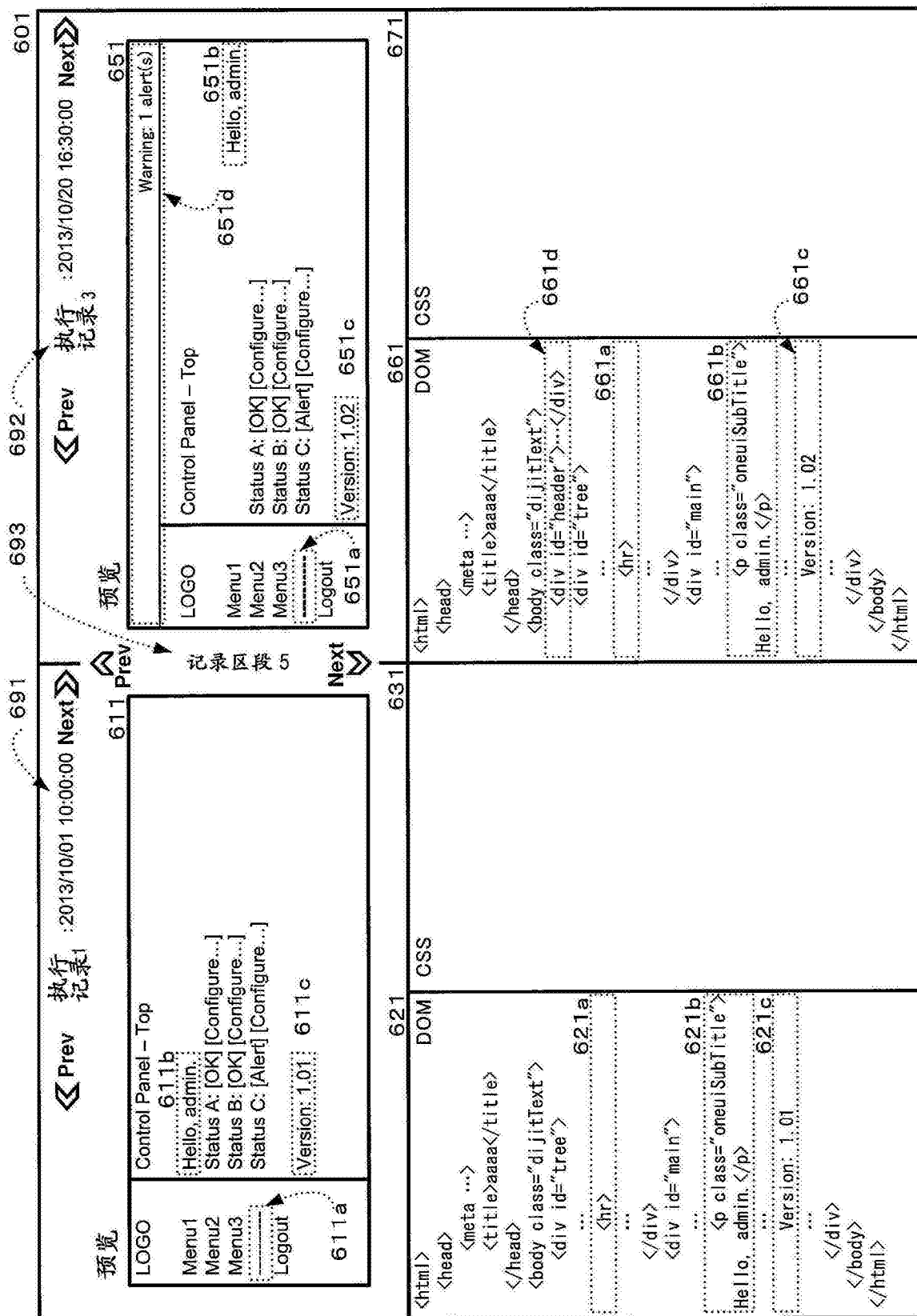


图6A

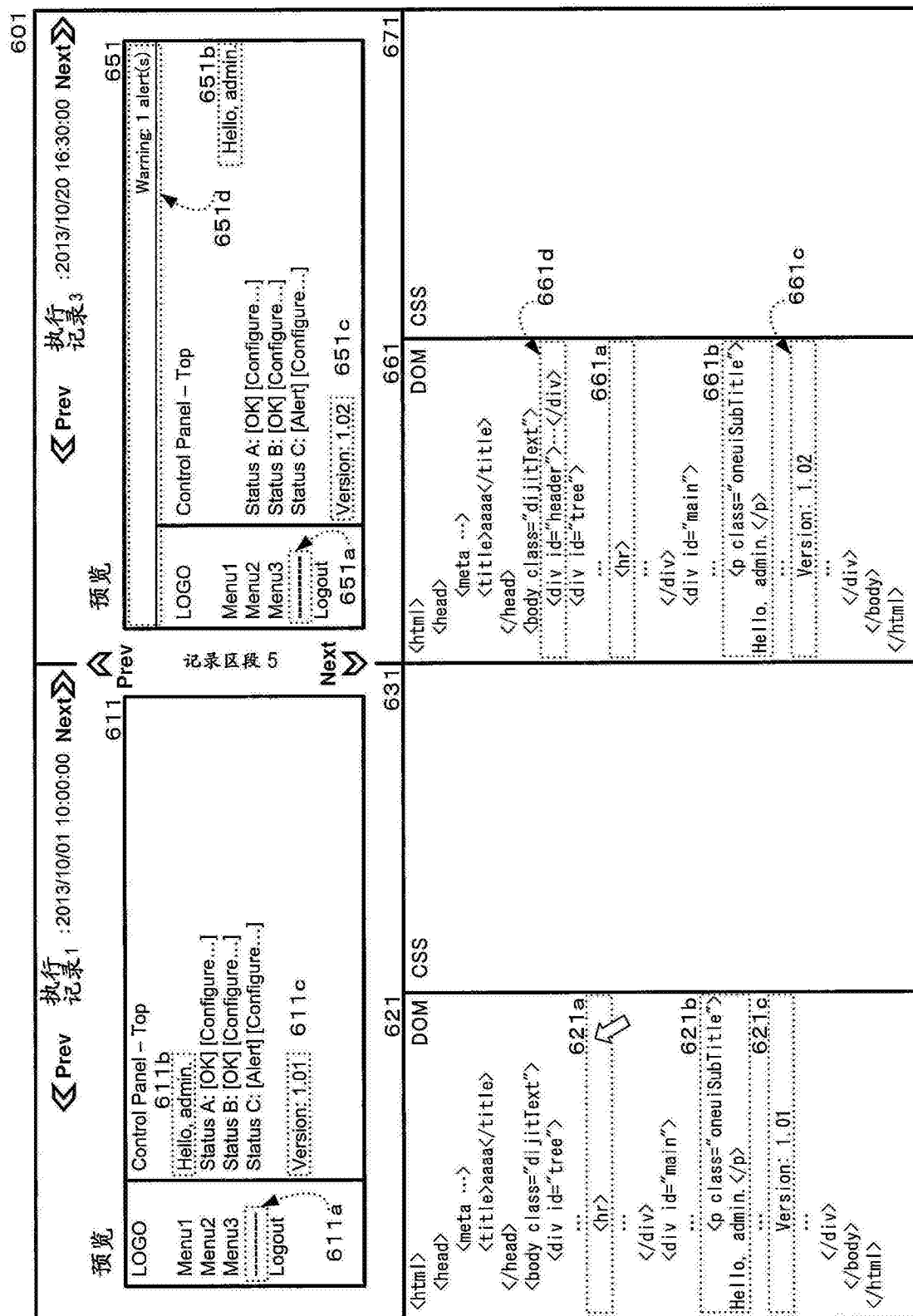


图6B

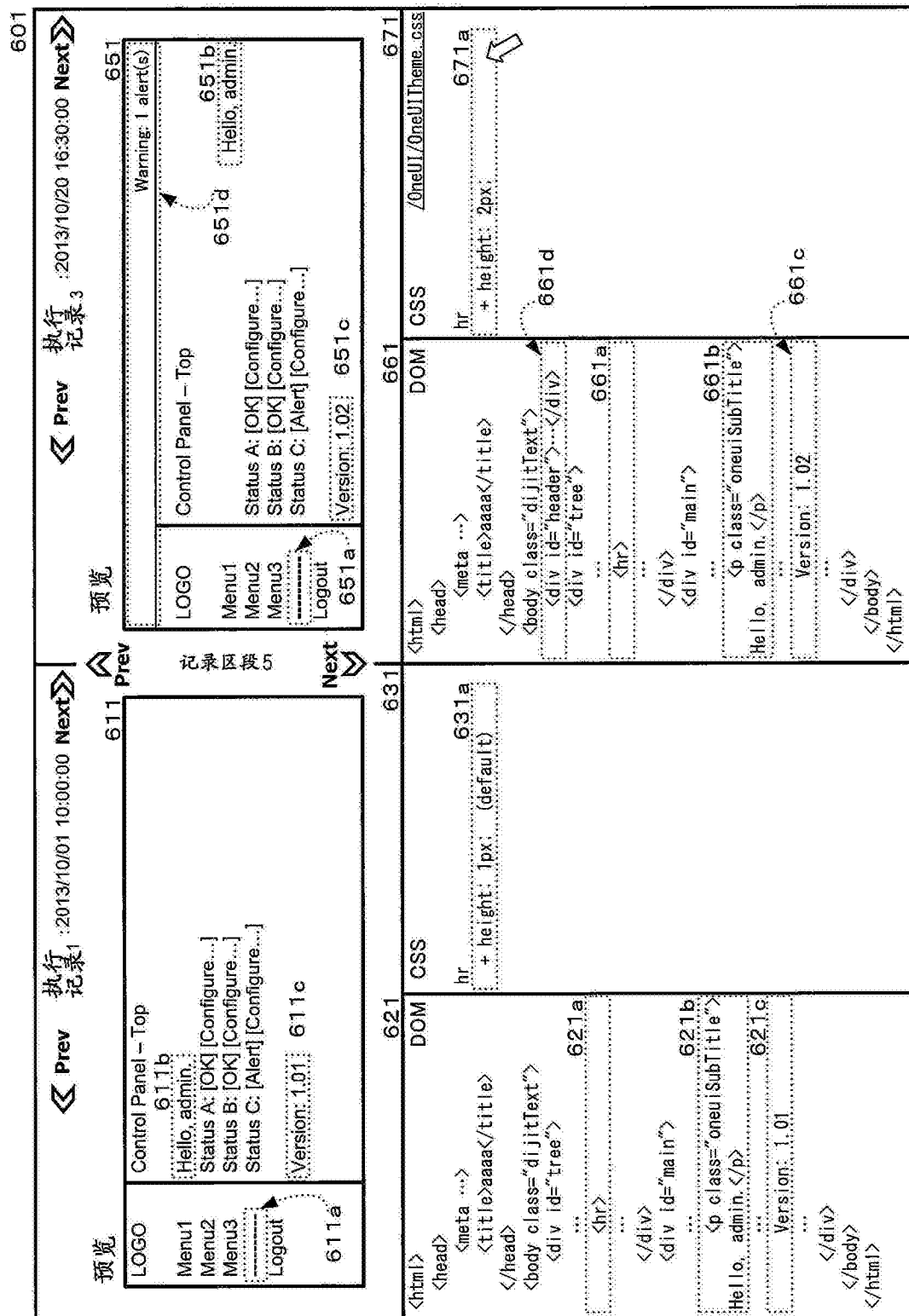


图6C

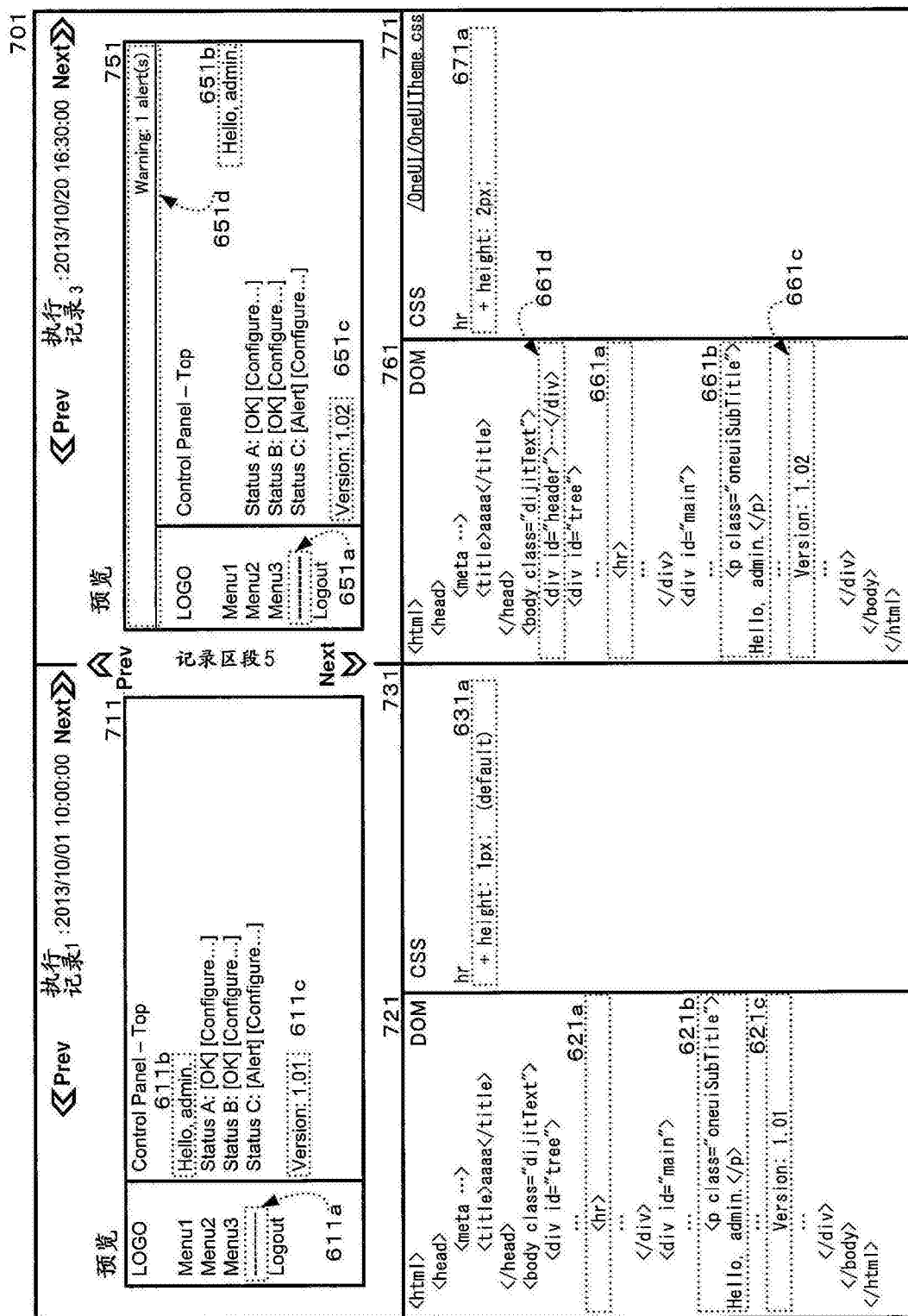


图7A

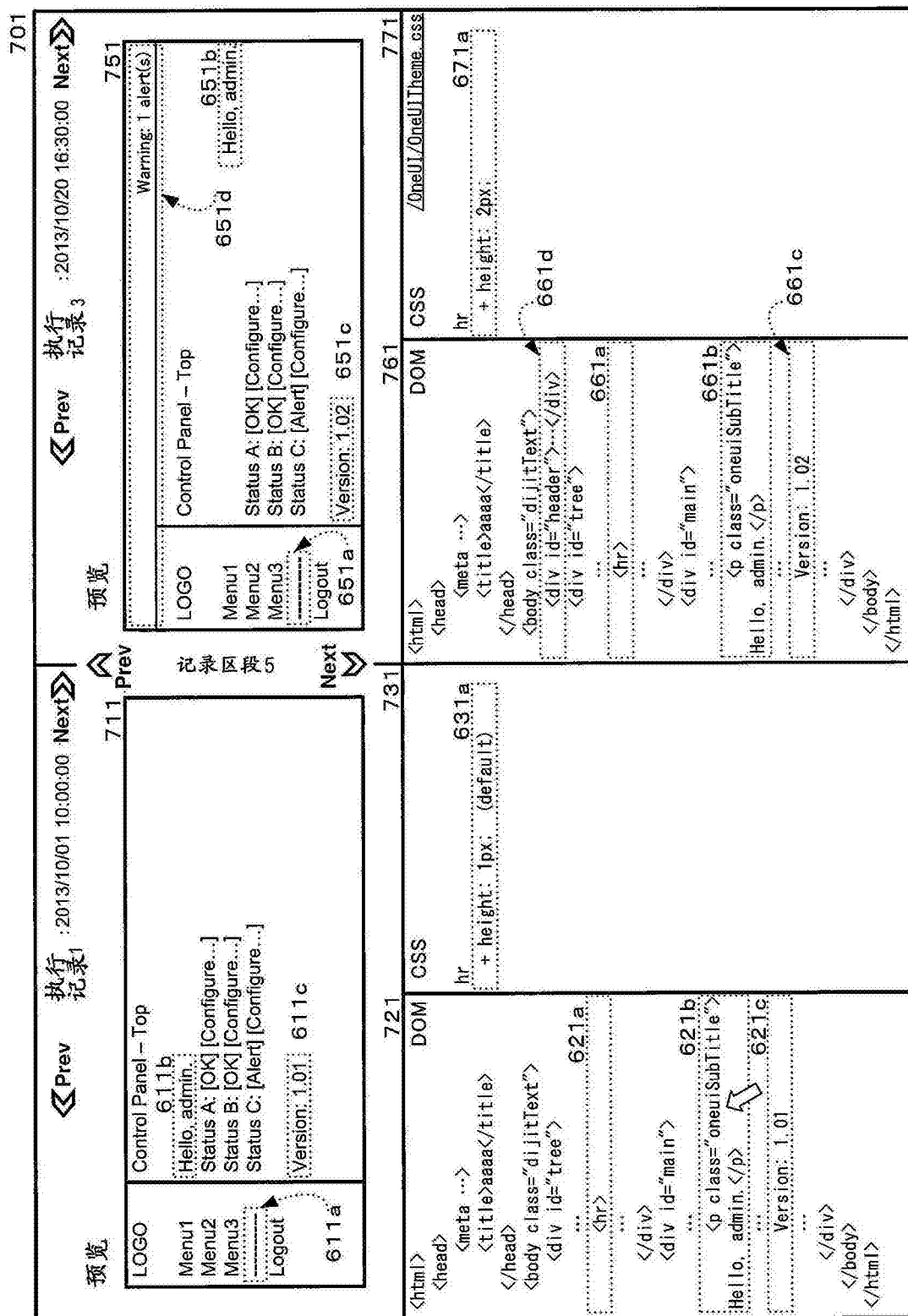


图7B

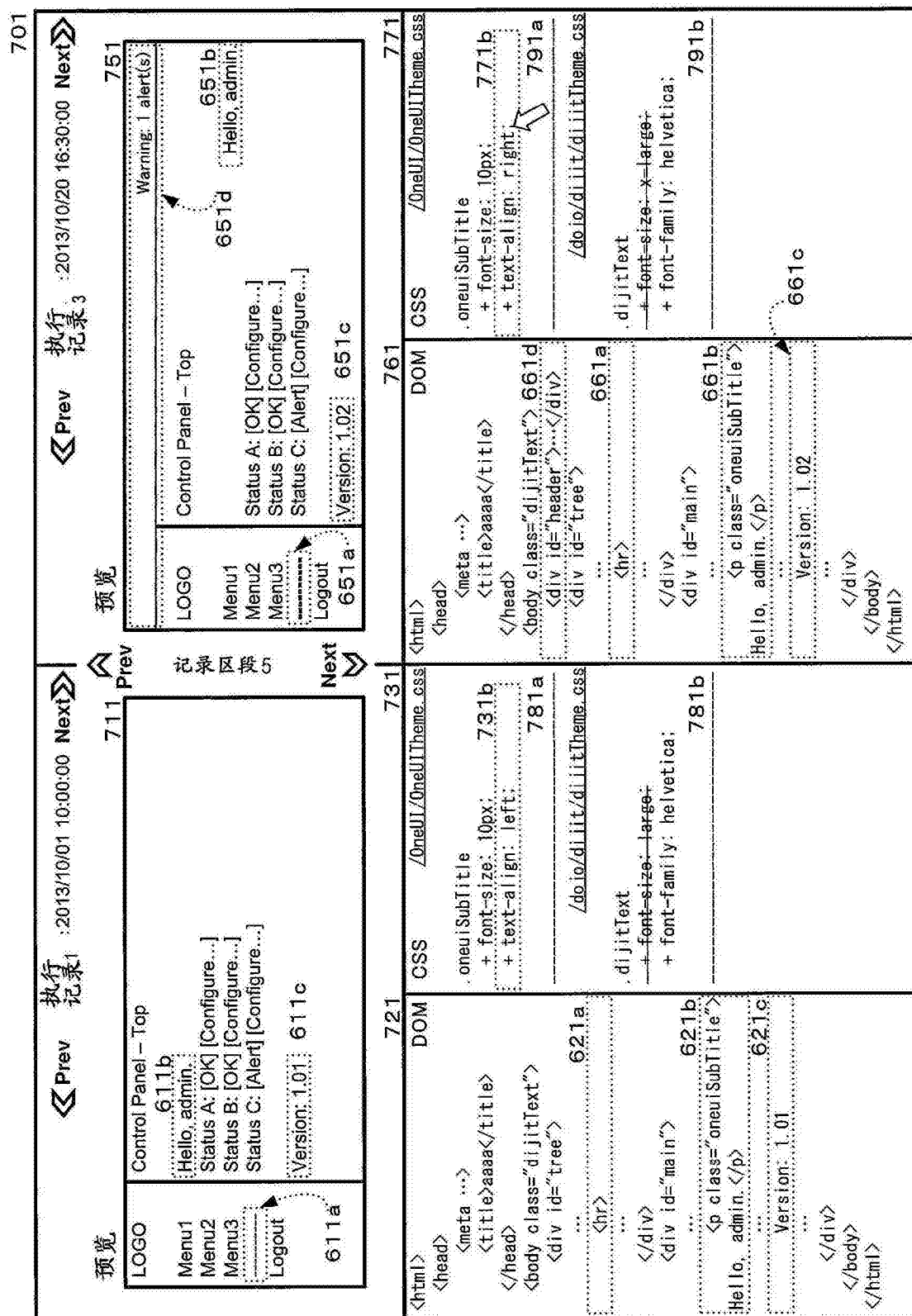


图7C

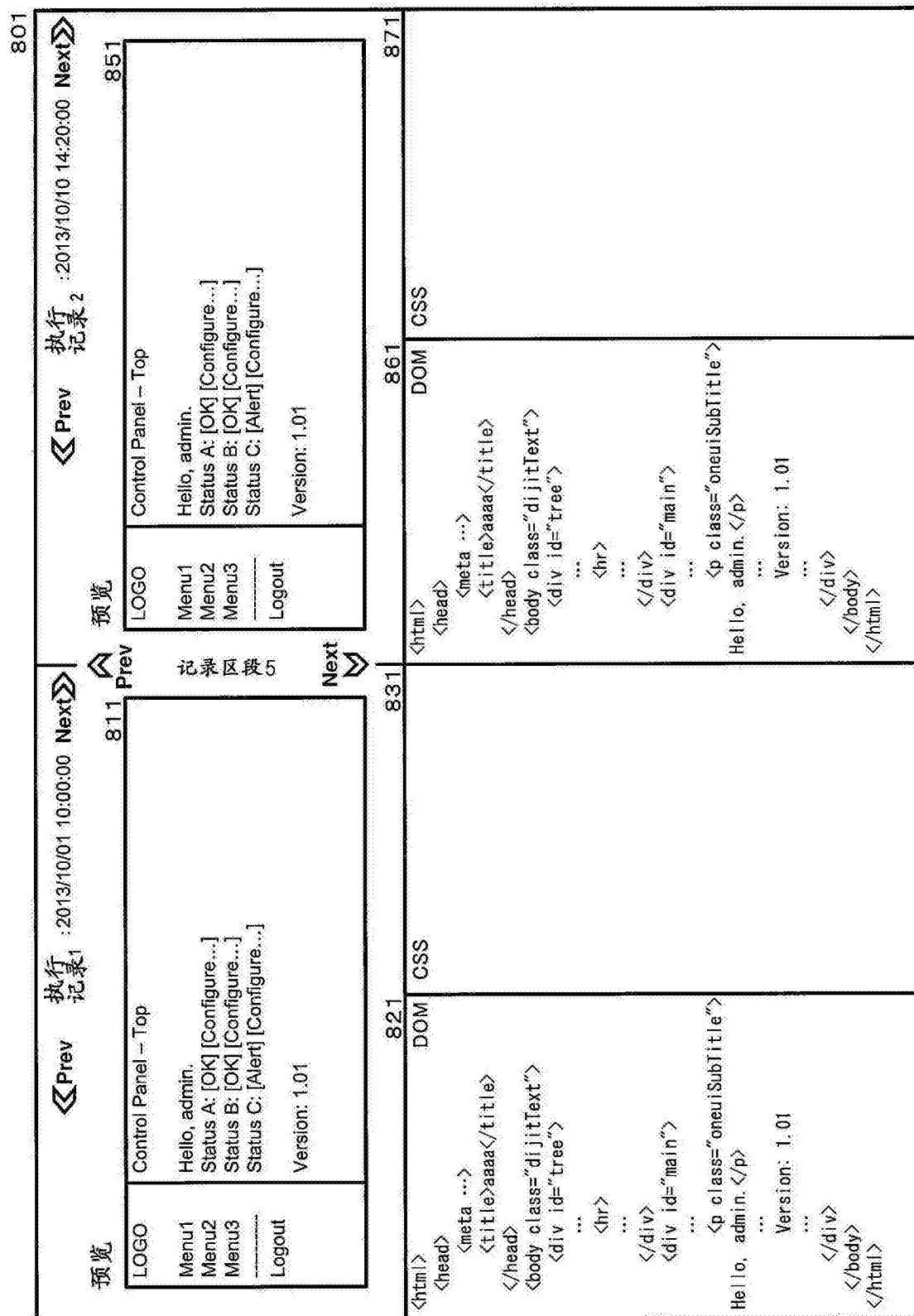
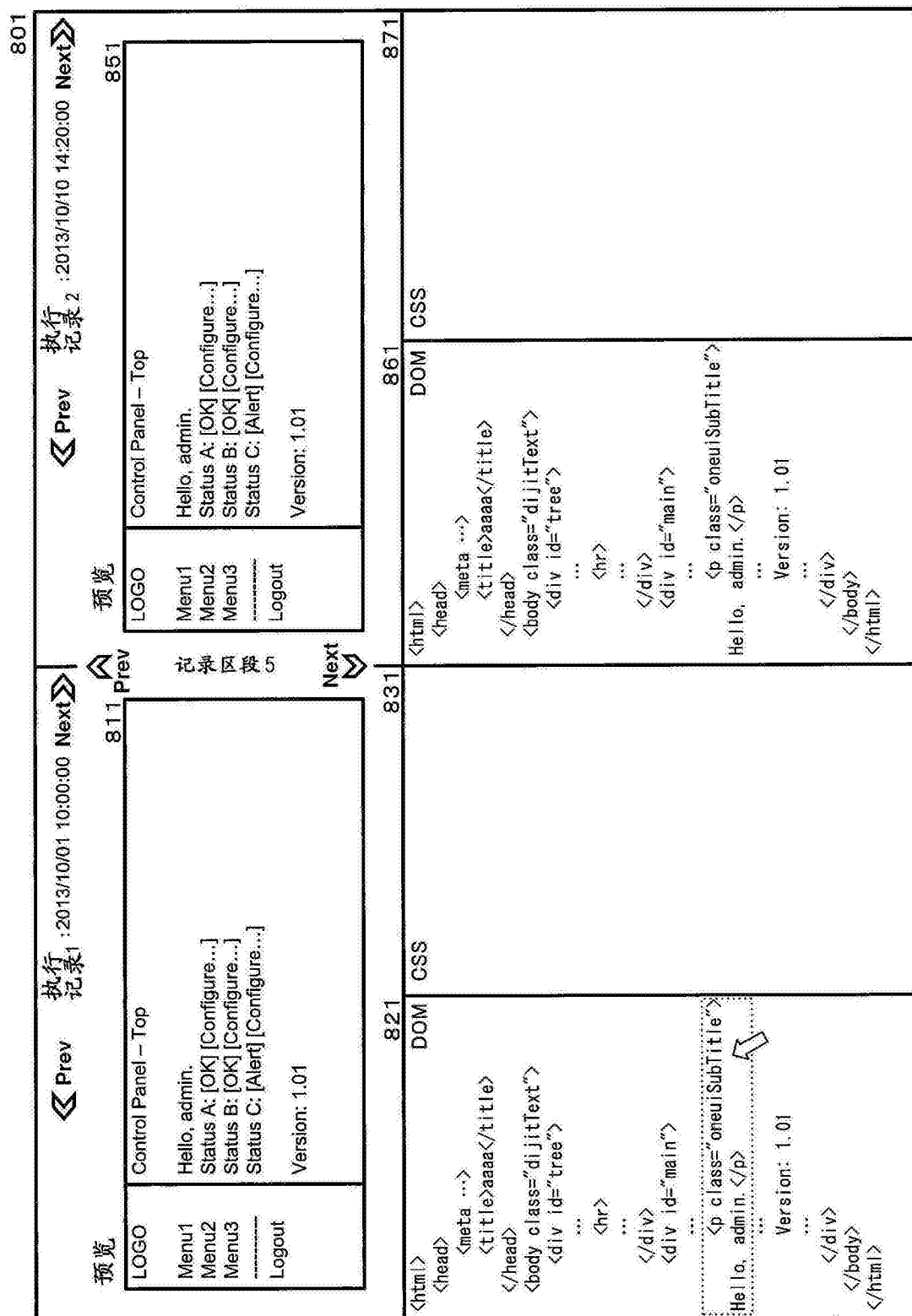


图 8A



801

执行记录2

2013/10/10 14:20:00

Next

Prev

预览

Control Panel - Top

Menu1
Menu2
Menu3

Logout

Hello, admin.
Status A: [OK] [Configure...]
Status B: [OK] [Configure...]
Status C: [Alert] [Configure...]
Version: 1.01

851

801

执行记录1

2013/10/01 10:00:00

Next

Prev

预览

Control Panel - Top

Menu1
Menu2
Menu3

Logout

801

执行记录2

2013/10/10 14:20:00

Next

Prev

预览

Control Panel - Top

Menu1
Menu2
Menu3

Logout

801

执行记录2

2013/10/10 14:20:00

Next

Prev

预览

Control Panel - Top

Menu1
Menu2
Menu3

Logout

801

执行记录2

2013/10/10 14:20:00

Next

Prev

预览

Control Panel - Top

Menu1
Menu2
Menu3

Logout

801

执行记录2

2013/10/10 14:20:00

Next

Prev

预览

Control Panel - Top

Menu1
Menu2
Menu3

Logout

801

执行记录2

2013/10/10 14:20:00

Next

Prev

预览

Control Panel - Top

Menu1
Menu2
Menu3

Logout

801

执行记录2

2013/10/10 14:20:00

Next

Prev

预览

Control Panel - Top

Menu1
Menu2
Menu3

Logout

801

执行记录2

2013/10/10 14:20:00

Next

Prev

预览

Control Panel - Top

Menu1
Menu2
Menu3

Logout

801

执行记录2

2013/10/10 14:20:00

Next

Prev

预览

Control Panel - Top

Menu1
Menu2
Menu3

Logout

801

执行记录2

2013/10/10 14:20:00

Next

Prev

预览

Control Panel - Top

Menu1
Menu2
Menu3

Logout

801

执行记录2

2013/10/10 14:20:00

Next

Prev

预览

Control Panel - Top

Menu1
Menu2
Menu3

Logout

801

执行记录2

2013/10/10 14:20:00

Next

Prev

预览

Control Panel - Top

Menu1
Menu2
Menu3

Logout

801

执行记录2

2013/10/10 14:20:00

Next

Prev

预览

Control Panel - Top

Menu1
Menu2
Menu3

Logout

801

执行记录2

2013/10/10 14:20:00

Next

Prev

预览

Control Panel - Top

Menu1
Menu2
Menu3

Logout

801

执行记录2

2013/10/10 14:20:00

Next

Prev

预览

Control Panel - Top

Menu1
Menu2
Menu3

Logout

801

执行记录2

2013/10/10 14:20:00

Next

Prev

预览

Control Panel - Top

Menu1
Menu2
Menu3

Logout

801

执行记录2

2013/10/10 14:20:00

Next

Prev

预览

Control Panel - Top

Menu1
Menu2
Menu3

Logout

801

执行记录2

2013/10/10 14:20:00

Next

Prev

预览

Control Panel - Top

Menu1
Menu2
Menu3

Logout

801

执行记录2

2013/10/10 14:20:00

Next

Prev

预览

Control Panel - Top

Menu1
Menu2
Menu3

Logout

801

执行记录2

2013/10/10 14:20:00

Next

Prev

预览

Control Panel - Top

Menu1
Menu2
Menu3

Logout

801

执行记录2

2013/10/10 14:20:00

Next

Prev

预览

Control Panel - Top

Menu1
Menu2
Menu3

Logout

801

执行记录2

2013/10/10 14:20:00

Next

Prev

预览

Control Panel - Top

Menu1
Menu2
Menu3

Logout

801

执行记录2

2013/10/10 14:20:00

Next

Prev

预览

Control Panel - Top

Menu1
Menu2
Menu3

Logout

801

执行记录2

2013/10/10 14:20:00

Next

Prev

预览

Control Panel - Top

Menu1
Menu2
Menu3

Logout

801

执行记录2

2013/10/10 14:20:00

Next

Prev

预览

Control Panel - Top

Menu1
Menu2
Menu3

Logout

801

执行记录2

2013/10/10 14:20:00

Next

Prev

预览

Control Panel - Top

Menu1
Menu2
Menu3

Logout

801

执行记录2

2013/10/10 14:20:00

Next

Prev

预览

Control Panel - Top

Menu1
Menu2
Menu3

Logout

801

执行记录2

2013/10/10 14:20:00

Next

Prev

预览

Control Panel - Top

Menu1
Menu2
Menu3

Logout

801

执行记录2

2013/10/10 14:20:00

Next

Prev

预览

Control Panel - Top

Menu1
Menu2
Menu3

Logout

801

执行记录2

2013/10/10 14:20:00

Next

Prev

预览

Control Panel - Top

Menu1
Menu2
Menu3

Logout

801

执行记录2

2013/10/10 14:20:00

Next

Prev

预览

Control Panel - Top

Menu1
Menu2
Menu3

Logout

801

执行记录2

2013/10/10 14:20:00

Next

Prev

预览

Control Panel - Top

Menu1
Menu2
Menu3

Logout

801

执行记录2

2013/10/10 14:20:00

Next

Prev

预览

Control Panel - Top

Menu1
Menu2
Menu3

Logout

801

执行记录2

2013/10/10 14:20:00

Next

Prev

预览

Control Panel - Top

Menu1
Menu2
Menu3

Logout

801

执行记录2

2013/10/10 14:20:00

Next

Prev

预览

Control Panel - Top

Menu1
Menu2
Menu3

Logout

801

执行记录2

2013/10/10 14:20:00

Next

Prev

预览

Control Panel - Top

Menu1
Menu2
Menu3

Logout

801

执行记录2

2013/10/10 14:20:00

Next

Prev

预览

Control Panel - Top

Menu1
Menu2
Menu3

Logout

801

执行记录2

2013/10/10 14:20:00

Next

Prev

预览

Control Panel - Top

Menu1
Menu2
Menu3

Logout

801

执行记录2

2013/10/10 14:20:00

Next

Prev

预览

Control Panel - Top

Menu1
Menu2
Menu3

Logout

801

执行记录2

2013/10/10 14:20:00

Next

Prev

预览

Control Panel - Top

Menu1
Menu2
Menu3

Logout

801

执行记录2

2013/10/10 14:20:00

Next

Prev

预览

Control Panel - Top

Menu1
Menu2
Menu3

Logout

801

执行记录2

2013/10/10 14:20:00

Next

Prev

预览

Control Panel - Top

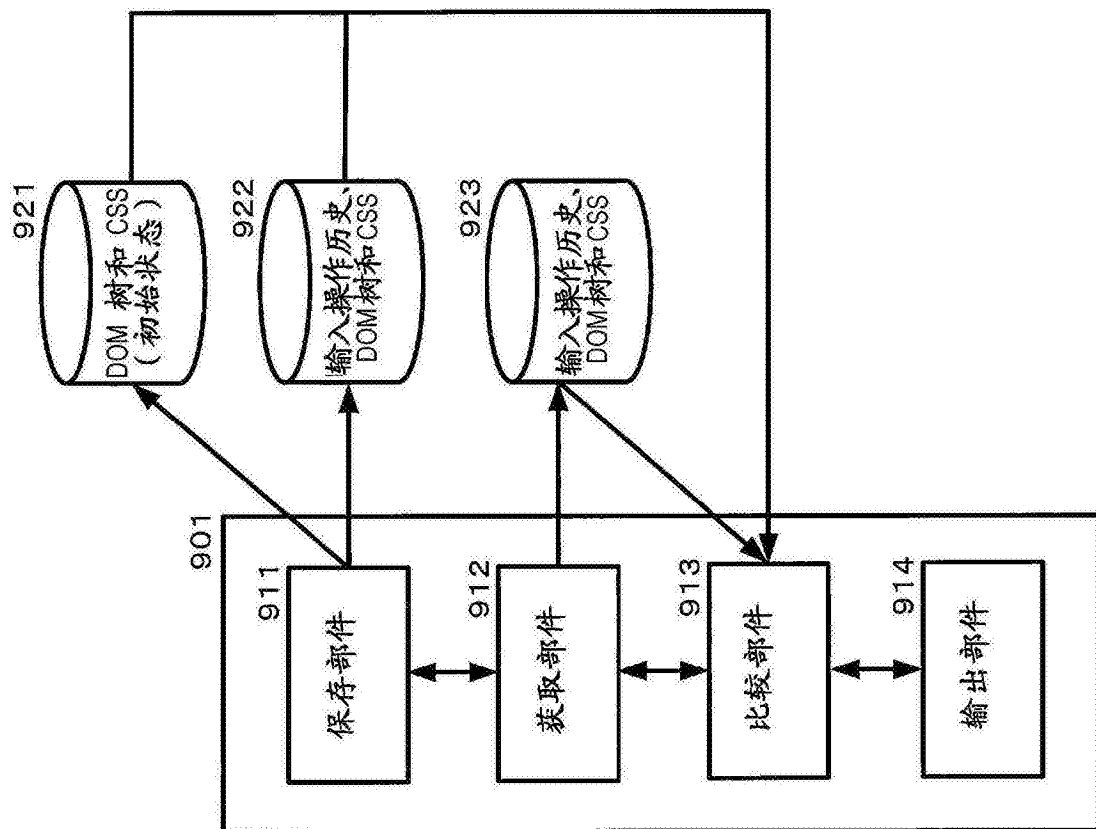


图9

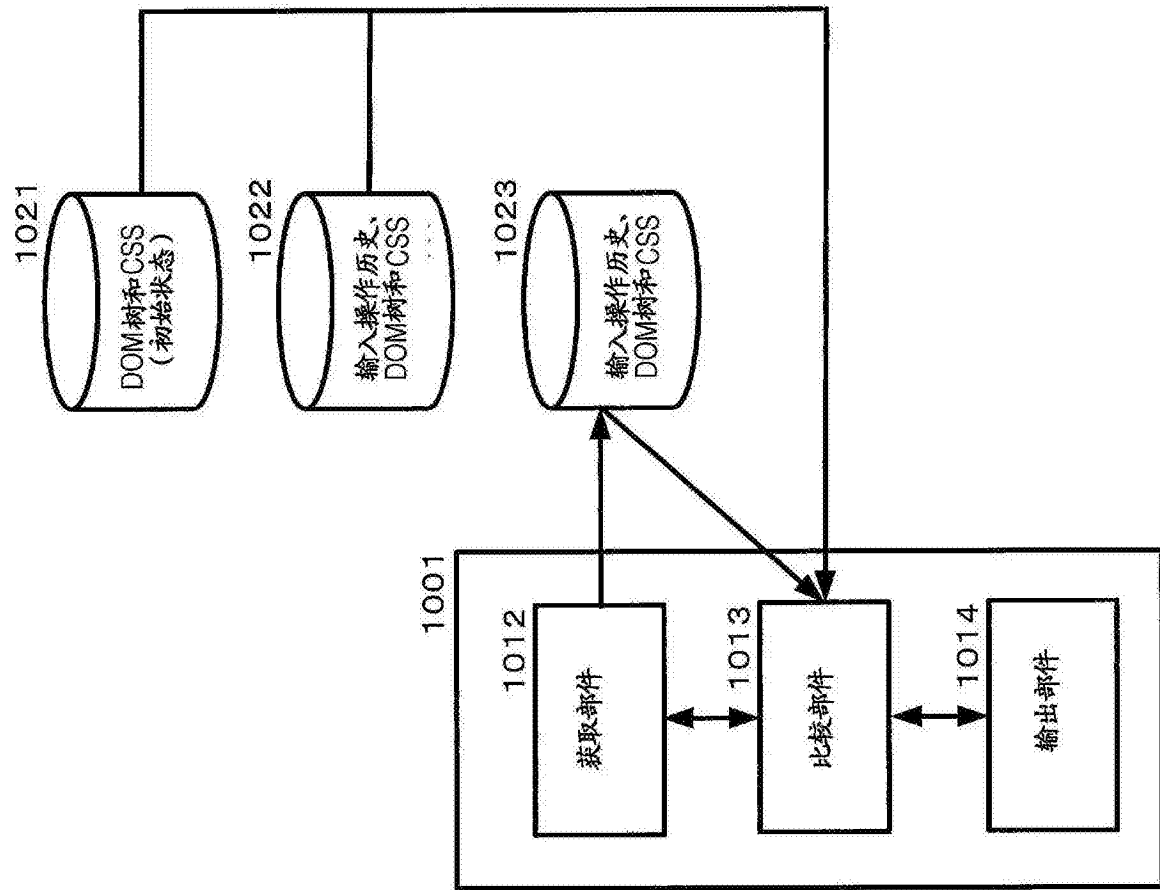


图10