(54) 发明名称
协同浏览与其他形式的信息共享的集成

(57) 摘要
一种协同浏览服务使用JavaScript来允许显示在用户浏览器的网页被远程查看。对网页的更新呈现在HTML中并转发到协同浏览会话上。在拖拽要素列表中规定了不应当可见的网页方面，从而防止JavaScript将这些要素的内容传输至协同浏览会话上。在远程位置查看该网页的人员可以选择对象，以使得这些对象在用户浏览器内突出显示。同样地，查看网页的人员可通过选择对象并在该对象内输入信息来操控对象。对网页的更新进行收集和汇聚，使得仅将最新的更新转发给协同浏览会话，不对DOM产生影响的更新例如悬停状态也被传输至会话。
1. 一种在第二位置查看第一浏览器的内容的方法，包括：
   使待提供的协同浏览脚本连同待加载至所述第一浏览器中的网页一起运行，其中所述协同浏览脚本配置为能使对所述第一浏览器的内容的更新呈现在所述第一浏览器的HTML中并能将所述更新转发给协同浏览服务，以使得所述第一浏览器的内容再现于所述第二位置处；
   通过协同浏览服务从所述第一浏览器接收更新；和
   将所述更新转发至所述第二位置。
2. 根据权利要求1所述的方法，其中，所述第一浏览器中所显示的内容由文档对象模型DOM来描述。
3. 根据权利要求1所述的方法，其中，从网络服务器将所述网页提供给所述浏览器，其中所述协同浏览脚本由网络服务器来提供。
4. 根据权利要求1所述的方法，其中，从网络服务器将所述网页提供给所述浏览器，其中所述协同浏览脚本由协同浏览服务来提供。
5. 根据权利要求1所述的方法，其中，如果所述网页在所述第一浏览器中不完全可见，那么在所述第二位置处仅有所述网页在所述第一浏览器中可见的相同部分是可见的。
6. 根据权利要求5所述的方法，其中，当所述第一浏览器滚动或所述第一浏览器调整尺寸时，所述第二位置处的网页视图自动进行更新。
7. 根据权利要求5所述的方法，其中，如果所述第一浏览器打开多个选项卡，那么所述第二位置仅能够看到所述选项卡中的一个选项卡，由此使得所述第一浏览器中具有焦点的那个选项卡在所述第二位置是可见的。
8. 根据权利要求1所述的方法，其中，来自所述第一浏览器的更新包括鼠标指针在所述第一浏览器中的位置，并且其中向所述第二位置转发所述更新的做法包括：向所述第二位置转发所述鼠标指针的位置，以使得所述第一浏览器的鼠标位置能够复制在所述第二位置上。
9. 根据权利要求1所述的方法，其中，如果显示在所述第一浏览器上的网页包括多个输入字段，那么来自所述第一浏览器的更新则包括关于所述多个输入字段中的一个字段当前在所述第一浏览器中是否有焦点的指示。
10. 根据权利要求1所述的方法，其中，所述第一浏览器的用户在网页上所采取的动作被反映为来自所述第一浏览器的更新。
11. 根据权利要求10所述的方法，其中，所述网页上所采取的动作包括：由用户点击项目以选择所述网页上的对象。
12. 根据权利要求1所述的方法，其中，协同浏览脚本JavaScript促使所述第一浏览器向所述协同浏览服务上传来自所述第一浏览器的网页HTML及其它数据；并且，所述所述协同浏览服务将此数据呈现为能在所述第二位置进行查看的网页。
13. 根据权利要求1所述的方法，其中，所述JavaScript是浏览器插件。
14. 根据权利要求1所述的方法，其中，通过所述协同浏览服务从所述第一浏览器接收更新的步骤以及向所述第二位置转发所述更新的步骤均发生在所述协同浏览服务主管的协同浏览会话中。
15. 根据权利要求14所述的方法，其中，一旦所述第二位置已加入会话，那么至少一个
权利要求书

16. 根据权利要求14所述的方法，其中，所述至少一个会话cookie是安全会话cookie，其包含要用所述协同浏览服务已知的密钥来签署的会话ID。

17. 根据权利要求14所述的方法，其中，所述至少一个会话cookie包括用于查看与所述协同浏览会话相关联的HTTP页面的安全cookie以及用于查看与所述协同浏览会话相关联的HTTP页面的非安全cookie。

18. 根据权利要求14所述的方法，还包括记录所述协同浏览会话。

19. 根据权利要求1所述的方法，其中，从所述第一浏览器接收更新的步骤包括：在用户将信息输入到所述第一浏览器中时从所述第一浏览器接收输入的步骤，转发所述更新的步骤包括：向所述第二位置转发所述用户输入。

20. 根据权利要求19所述的方法，其中，用户输入包括：由用户选择按钮、选择复选框，或者将至少键入输入到可选字段中。

21. 一种在第二位置查看第一浏览器的内容的方法，包括：
   使待提供的协同浏览脚本连同待加载到所述第一浏览器中的网页一起运行，其中所述协同浏览脚本配置为能使对所述第一浏览器的内容的更新呈现在所述第一浏览器端的HTML中并能将所述更新转发至第二位置，以使得所述第一浏览器的内容再现于所述第二位置处；
   选择与所述第二位置处再现的内容相关联的对象；和
   使所述选择的对象在所述第一浏览器端被可视地识别。

22. 根据权利要求21所述的方法，其中，使所述选择的对象在所述第一浏览器端被可视地识别的做法包括：使所述第一浏览器的网页中的对应对象突出显示。

23. 根据权利要求21所述的方法，还包括在第二位置处对所述选择的对象进行可视识别。

24. 根据权利要求23所述的方法，还包括在所述第一浏览器端停止对所述选择的对象进行可视识别。

25. 根据权利要求24所述的方法，其中，在预设定时之后自动停止所述选择的对象被可视识别的步骤。

26. 根据权利要求24所述的方法，其中，当在所述第二位置处选择第二对象时停止所述选择的对象被可视识别的步骤。

27. 根据权利要求24所述的方法，其中，当在所述第二位置处第二次选择所述对象时停止所述选择的对象被可视识别的步骤。

28. 根据权利要求23所述的方法，其中，使所述选择的对象在第二位置处被可视地识别的做法包括：在所述第一浏览器端和所述第二位置处共同绘制围绕所述对象的视觉可区分框。

29. 根据权利要求21所述的方法，其中，选择与所述第二位置处再现的内容相关联的对象的做法使运行在所述第二位置上的脚本产生能唯一识别出所述对象的CSS选择器；所述方法还包括将所述CSS选择器传递给所述第一浏览器端的JavaScript的步骤。

30. 一种在第二位置选择性地不让特定内容在第一浏览器的外部可见的方法，包括：
   使待提供的协同浏览脚本连同待加载到所述第一浏览器中的网页一起运行，其中所
权利要求书

述协同浏览脚本配置为能使对所述第一浏览器的内容的更新呈现在所述第一浏览器端的 HTML 中并能将所述更新转发给协同浏览服务，以使得所述第一浏览器的内容再现于所述第二位置处；

使隐藏要素列表与所述协同浏览脚本一起提供，其中所述隐藏要素列表在第二位置处识别那些内容在所述第一浏览器外部为不可见的网页要素。

31. 根据权利要求 30 所述的方法，还包括在所述第二位置接收来自所述第一浏览器的更新。

32. 根据权利要求 31 所述的方法，其中，在所述第二位置处接收的更新包含加载到所述第一浏览器中的所有网页要素的信息，其中包括至少一些所述隐藏要素，并且其中所述隐藏要素的内容在从所述更新接收到的信息中是无法辨别的。

33. 根据权利要求 31 所述的方法，其中，所述隐藏要素的内容不通过协同浏览脚本传输，从而使得所述隐藏要素的内容不离开所述第一浏览器。

34. 根据权利要求 30 所述的方法，还包括在所述第一浏览器中识别所述隐藏要素，以通知所述第一浏览器的用户；还包括在这些字段中的信息不会在网络上上传输。

35. 一种从第二位置选择性地启用对第一浏览器的远程控制的方法，包括:

使待提供的协同浏览脚本连同加载到所述第一浏览器中的网页一起运行，其中所述协同浏览脚本配置为能使得所述第一浏览器的内容的更新呈现在所述第一浏览器端的 HTML 中并能将所述更新转发给协同浏览服务，以使得所述第一浏览器的内容再现于所述第二位置处；

从所述第一浏览器接收更新；和

捕捉所述第二位置处的用户输入，并将来自所述第二位置处的用户输入作为输入转发给所述第一浏览器，从而控制所述第一浏览器的操作。

36. 根据权利要求 35 所述的方法，其中，所述第二位置处的用户输入包括键盘敲击、鼠标移动和鼠标点击。

37. 根据权利要求 35 所述的方法，使隐藏对象列表连同所述协同浏览脚本一起提供，其中所述隐藏对象列表识别那些不能被所述第二位置处的用户输入所控制的网页对象。

38. 根据权利要求 37 所述的方法，其中，所述隐藏对象列表指定了所述网页的表格、div 和页面。

39. 根据权利要求 37 所述的方法，其中，所述隐藏对象列表防止所述第二位置处的用户点击所述网页上的链接或按钮。

40. 根据权利要求 35 所述的方法，还包括通过所述第一浏览器捕捉来自所述第二位置的输入来作为所述第一浏览器的相关字段中的输入，从而为所述第二位置处提供对所述第一浏览器的操作方面的直接控制。

41. 一种将信息从第二位置提供到第一浏览器中的方法，包括:

使待提供的协同浏览脚本连同加载到所述第一浏览器中的网页一起运行，其中所述协同浏览脚本配置为能使得所述第一浏览器的内容的更新呈现在所述第一浏览器端的 HTML 中并能将所述更新转发给协同浏览服务，以使得所述第一浏览器的内容再现于所述第二位置处；

开启协同浏览会话，其中将所述网页视图的更新提供并显示在所述第二位置上；和
权利要求书

42. 根据权利要求 41 所述的方法，其中，提供到所述第一浏览器中的信息提供在所述第一个浏览器内的窗口中。

43. 根据权利要求 42 所述的方法，其中，从所述第二位置提供到所述第一浏览器中的信息是与所述第二位置相关联的计算机显示器的视图。

44. 根据权利要求 42 所述的方法，其中，从所述第二位置提供到所述第一浏览器中的信息是来自所述第二位置的相机的实况视频。

45. 根据权利要求 42 所述的方法，其中，从所述第二位置提供到所述第一浏览器中的信息是来自连接在所述第二位置的视频源的实况视频。

46. 根据权利要求 41 所述的方法，其中，提供到所述第一浏览器中的信息是针对下载屏幕共享软件的提示，其中所述屏幕共享软件用于捕捉那些显示在能查看所述第一浏览器的显示器上的信息。

47. 根据权利要求 41 所述的方法，其中，提供到所述第一浏览器中的信息是用于在与所述第一浏览器相关联的第一计算机上运行屏幕共享客户端以发起屏幕共享会话的呼叫，其中在屏幕共享会话上，捕捉与所述第一计算机相关联的显示器上所显示的信息并将其传输至所述第二位置。

48. 根据权利要求 47 所述的方法，还包括由所述第二位置自动启用对所述屏幕共享会话的远程控制的步骤。

49. 一种从第一浏览器向第二位置提供信息的方法，包括：

使待提供的协同浏览器脚本连同加载到所述第一浏览器中的网页一起运行，其中所述协同浏览器脚本配置为能使对所述第一浏览器的内容的更新呈现在所述第一浏览器端的 HTML 中并能将所述更新转发给协同浏览器，以使得所述第一浏览器的内容再现于所述第二位置的；

检测对所述第一浏览器所显示的网页的更新，并将所述更新转发给所述第二位置；和

将所述动作的指示转发给所述第二位置。

50. 根据权利要求 49 所述的方法，其中，可暂停要素是当鼠标暂停在它上方时而具有适用于它或它的同族或子族的样式的要素。

51. 根据权利要求 49 所述的方法，还包括由所述第一浏览器将可暂停要素上方的鼠标位置解读为被暂停要素上的暂停动作的步骤；以及，由所述第一浏览器确定；第二位置浏览器中的代理 JavaScript 定义何种样式属性应用于所述可暂停要素，以在代理端上模拟暂停。

52. 根据权利要求 50 所述的方法，其中，如果被暂停要素上方的暂停动作不影响文档对象模型 DOM 定义所述网页的结构，那么所述方法还包括向所述第二位置转发所述被暂停要素的身份。

53. 根据权利要求 48 所述的方法，还包括为所述第二位置提供可暂停要素列表的步骤。

54. 根据权利要求 53 所述的方法，其中，所述可暂停要素列表是针对所述网站的 CSS 选
择器。

55. 根据权利要求 54 所述的方法，其中，所述可跨停要素列表与要应用到所述第二位置的网站视图的 CSS 样式集合相关联。

56. 根据权利要求 53 所述的方法，其中，从有助于所述第一浏览器与所述第二位置之间的信息交换的协同浏览服务中检查所述可跨停要素列表。

57. 根据权利要求 56 所述的方法，其中，通过监控所述第一浏览器端的鼠标位置来复制所述第一浏览器的悬停状态，并在所述第二位置选择性地重新创建所述悬停状态。

58. 根据权利要求 57 所述的方法，其中，所述第二位置不运行来自网页的脚本，而是显示从所述第一浏览器接收的 HTML，并且其中所述第二位置不能检测与所述第一浏览器的鼠标位置相关联的悬停状态。

59. 一种从第一浏览器向第二位置提供信息的方法，包括：
接收协同浏览脚本，所述协同浏览脚本配置为连同要加载到所述第一浏览器中的网页一起运行，其中所述协同浏览脚本能使对所述第一浏览器的内容的更新呈现在所述第一浏览器端的 HTML 中，并将所述更新转发给协同浏览器服务，以使得所述第一浏览器的内容再现于所述第二位置处；
产生包含所述第一浏览器在第一时间点上的完整视图的完整更新；
检测与所述网页相关的文档对象模型 DOM 的变化；和
基于 DOM 的变化来产生更新。

60. 根据权利要求 59 所述的方法，其中，所述更新是由产生所述完整更新的步骤开始基于所有变化的累积更新。

61. 根据权利要求 60 所述的方法，其中，与给定对象相关的 DOM 的连续变化将导致连续累积更新的产生，从而使在任何时间点针对所述给定对象存在至多一个更新。

62. 根据权利要求 60 所述的方法，其中，生成更新的步骤包括：评估 DOM 中的要素的连续实例之间的差异，并对 DOM 中的每个要素进行递归。

63. 根据权利要求 62 所述的方法，其中，评估 DOM 中的要素的连续实例之间的差异的步骤包括以下步骤：
确定所述要素的属性是否已改变；和
确定所述要素的子列表是否已改变。

64. 根据权利要求 63 所述的方法，其中，如果所述要素的子列表发生变化，那么所述方法还包括将 HTML 差异要素添加到该更新版要素中包含 HTML 的更新中的步骤。

65. 根据权利要求 64 所述的方法，其中，如果所述子列表包含零个子要素，那么所述方法包括确定所述要素的文本是否已改变的步骤。

66. 根据权利要求 65 所述的方法，其中，如果所述要素的文本已改变，那么该方法还包括将 CONTENT 内容差异要素添加到该更新版要素中包含文本的更新中的步骤。

67. 根据权利要求 64 所述的方法，其中，如果所述子列表包含一个或多个子要素，那么所述方法包括比较每个子节点的数量、类型和节点值的步骤。

68. 根据权利要求 67 所述的方法，其中，如果在所述一个或多个子节点的数量、类型和节点值中检测到差异，那么所述方法还包括将 HTML 差异要素添加到该更新版要素中包含 HTML 的更新中的步骤。
69. 一种从通过协同浏览连接服务器维持的多个协同浏览会话中选择出与网站的访问者相关联的协同浏览会话的方法，包括：
识别与访问者相关联的浏览器；
和
正确识别出所述多个协同浏览会话中的哪个会话与所识别的浏览器相关联。

70. 根据权利要求 69 所述的方法，其中，识别所述浏览器包括：指示访问者通过点击显示在所述浏览器的网页上的按钮以向所述协同浏览连接服务器发送信号。

71. 根据权利要求 70 所述的方法，其中，点击所述按钮的做法为访问者提供了随机会话 ID，所述方法还包括以下步骤：从所述访问者接收所述随机会话 ID，并在上述正确识别出与所识别的浏览器相关联的协同浏览会话的步骤中使用所述随机会话 ID。

72. 根据权利要求 71 所述的方法，其中，上述正确识别出与所识别的浏览器相关联的协同浏览会话的步骤包括：由代理将所述随机会话 ID 提供给所述协同浏览连接服务器。

73. 根据权利要求 71 所述的方法，其中，上述正确识别出与所识别的浏览器相关联的协同浏览会话的步骤包括：通过代理利用所接收的随机会话 ID 从协同浏览会话列表中进行选择，以从中选择协同浏览会话。

74. 根据权利要求 69 所述的方法，其中，识别与访问者相关联的浏览器的做法包括：使所述访问者输入与发起协同浏览会话有关的识别信息，并使用所述识别信息来从多个协同浏览会话中选择所述协同浏览会话。

75. 根据权利要求 74 所述的方法，其中，所述识别信息包括所述访问者的个人信息，这包括访问者名称、账户 ID 或电子邮件地址。

76. 根据权利要求 69 所述的方法，还包括运行客户关系管理 CRM 软件，并且其中正确识别出协同浏览器会话中的哪个会话与所识别的浏览器相关联的步骤包括：由所述 CRM 软件选择针对所述访问者的记录。

77. 根据权利要求 76 所述的方法，还包括使得访问者输入与发起所述协同浏览器会话相关的识别信息，并且其中响应于所述访问者已输入识别信息的检测来实施上述由所述 CRM 软件选择记录的步骤。

78. 根据权利要求 77 所述的方法，其中，上述选择记录的步骤使得经由 CRM 软件向所述代理传送通知。

79. 根据权利要求 77 所述的方法，还包括：当所述访问者输入与发起协同浏览器会话有关的识别信息时，如果所述 CRM 软件尚不具有所述访问者的记录，则为访问者创建新的 CRM 软件记录。

80. 根据权利要求 76 所述的方法，其中，所述 CRM 软件具有显示所述网站的当前访问者的列表。

81. 根据权利要求 76 所述的方法，其中，针对所述访问者的 CRM 软件记录包含能够使所述代理加入与关联于所述 CRM 软件记录的访问者进行的协同浏览器会话的按钮。

82. 根据权利要求 69 所述的方法，还包括聊天服务，其为所述访问者和所述代理提供参与到基于文本或语音的聊天对话中的能力。

83. 根据权利要求 82 所述的方法，还包括：在发起所述代理与所述访问者之间的聊天会话的步骤之前，使所述访问者完成预聊调查的步骤。

84. 根据权利要求 83 所述的方法，还包括使用来自所述预聊调查的信息来识别与所述
访问者相关联的协同浏览会话的步骤。

85. 根据权利要求 84 所述的方法，其中，使用信息的步骤包括：在所述 CRM 软件系统中查找所述访问者的信息。

86. 根据权利要求 69 所述的方法，其中，所述网站包含能启动聊天服务的聊天服务脚本，从而发起所述代理和所述访问者之间的聊天会话。

87. 根据权利要求 86 所述的方法，其中，所述协同浏览脚本包括所述聊天服务脚本。

88. 根据权利要求 87 所述的方法，其中，所述协同浏览脚本和聊天服务脚本均是从所述聊天服务下载至所述浏览器。

89. 根据权利要求 86 所述的方法，其中，识别与所述访问者相关联的浏览器的步骤由所述聊天服务脚本来实现。

90. 根据权利要求 86 所述的方法，其中，所述正确识别出所述多个协同浏览会话中的哪个会话与所述的浏览器相关联的步骤是通过经由所述聊天服务脚从所述访问者所获得的信息来实现的。

91. 根据权利要求 90 所述的方法，还包括：在承担与所述访问者聊天会话的责任之后通过所述代理来正确识别的协同浏览会话来发起协同浏览会话的步骤。

92. 根据权利要求 91 所述的方法，其中，由所述代理发起协同浏览会话的步骤在建立协同浏览会话之前不需要所述访问者采取行动或提供许可。

93. 根据权利要求 92 所述的方法，其中，所述访问者不知道所述代理已发起协同浏览会话。

94. 根据权利要求 86 所述的方法，其中，所述协同浏览脚本包括唯一与所述代理和访问者之间的聊天会话相关联的协同浏览会话识别值。

95. 根据权利要求 94 所述的方法，其中，所述识别值使用在上述正确识别出所述多个协同浏览会话中的哪个会话与所述的浏览器相关联的步骤中，以便加入与所述访问者的协同浏览会话中。

96. 一种能使用户加入到与所述的访问者相关联的协同浏览会话的方法，所述方法包括：

由所述浏览器采用配置为在第一浏览器中运行的协同浏览脚本来发起协同浏览会话，其中所述协同浏览脚本配置为能使用对所述第一浏览器的内容的更新呈现在所述第一浏览器端的 HTML 中并能将所述更新转发给协同浏览服务，以使得所述第一浏览器的内容再现于第二位置处；

提供针对所述协同浏览会话的协同浏览访问链接，所述链接通过所述协同浏览服务可用于连接到所述协同浏览会话。

97. 根据权利要求 96 所述的方法，其中，提供链接的步骤包括：将针对所述协同浏览会话的协同浏览访问链接发布到社交媒体系统网站上。

98. 根据权利要求 97 所述的方法，其中，将针对所述协同浏览会话的协同浏览访问链接发布到与所述社交媒体系统联的访问者相关联的馈送器。

99. 根据权利要求 96 所述的方法，其中，发起协同浏览会话的步骤包括：由所述访问者选择网页上的协同浏览按钮，以发起对所述协同浏览脚本的下载和运行。

100. 根据权利要求 99 所述的方法，其中，无需所述访问者采取进一步行动，就能执行
提供链接的步骤。

101. 根据权利要求96所述的方法，还包括通知访问者其在所述社交媒体系统内的联系人也在所述网站上的步骤，且其中提供针对协同时机的会话的协同时机浏览访问链接的步骤包括：向同样出现在所述网站上的所述访问者在所述社交媒体系统内的联系人传输所述协同时机浏览访问链接。

102. 根据权利要求96所述的方法，其中，针对协同时机的会话的协同时机浏览访问链接包含会话标识符和关于主端所述协同时机的会话的所述协同时机浏览服务器的信息。

103. 根据权利要求96所述的方法，其中，提供针对协同时机的会话的协同时机浏览访问链接的步骤包括：通过即时消息或电子邮件来传输协同时机的会话的所述协同时机浏览访问链接。

104. 一种将屏幕共享会话与协同时机的会话集成的方法，该方法包括：在第一浏览器和第二位置之间建立协同时机会话；和在所述第二位置和所述第一浏览器之间建立屏幕共享会话；其中，所述屏幕共享会话出现的所述第一浏览器中。

105. 根据权利要求104所述的方法，其中，所述屏幕共享会话包括所述第二位置的文档相机中的视频馈送。

106. 根据权利要求104所述的方法，其中，所述屏幕共享会话包括所述第二位置的便携式手持装置中的视频馈送。

107. 根据权利要求104所述的方法，其中，所述屏幕共享会话包括所述第二位置的人员的摄像视图的视频馈送。

108. 根据权利要求104所述的方法，其中，所述屏幕共享会话包括显示在所述第二位置处的计算机监控器上的信息。

109. 根据权利要求104所述的方法，其中，所述屏幕共享会话选自：来自所述第二位置处的一个或多个外部源的一个或多个视频馈送之一，和显示在所述第二位置处的计算机监控器上的信息。

110. 根据权利要求109所述的方法，其中，包括所述屏幕共享会话中所述第二位置处的信息源无需结束所述屏幕共享会话就能改变。

111. 根据权利要求109所述的方法，其中，所述屏幕共享会话包括来自所述第二位置处的外部源的多个视频馈送。

112. 根据权利要求111所述的方法，其中，所述多个视频馈送中的至少一个馈送是在所述第二位置处的人员的摄像视图的实时视频馈送。

113. 根据权利要求104所述的方法，还包括：将所述第二位置作为访客加入所述屏幕共享会话，以使得所述第二位置处的人员能够查看正经由所述屏幕共享会话提供给所述第一浏览器的内容，从而使得所述第二位置即是将信息传输到所述屏幕共享会话上的信息源又是用于接收传输到所述屏幕共享会话上的内容的访客。

114. 一种对网站的访问者与在所述网站提供客户支持的代理之间的客户联系进行升级的方法，该方法包括：在所述访问者和所述代理之间发起基于计算机的聊天会话；发起基于计算机的协同时机浏览会话，同时与聊天会话合并，以使得所述代理能够从远程
位置查看客户浏览器的内容：和

由所述代理建立基于计算机的屏幕共享会话，从而通过协同浏览器会话将来自远程位置的信息源回显给所述访问者。

115. 根据权利要求 114 所述的方法，其中，所述协同浏览器会话使用 JavaScript 来建立，并且其中所述屏幕共享会话使用 AJAX 来建立。

116. 一种从第二位置查看正显示在用户浏览器上的浏览器会话内容的方法，所述方法包括：

当用户进入网站时，无需得知用户就能自动启动协同浏览器会话；

使第二用户从远程位置加入所述协同浏览器会话，从而使无需通知用户就能监控用户对所述网站的使用。

117. 根据权利要求 116 所述的方法，其中，启动协同浏览器会话的步骤包括：使所述用户浏览器定期地轮询连接服务器，以确定是否已发起所述协同浏览器会话。

118. 根据权利要求 117 所述的方法，其中，使第二用户加入协同浏览器会话的步骤包括：向所述连接服务器发送指令，以指示所述连接服务器来通知访问者 JavaScript 开始传输描述了用户浏览器的内容的更新。

119. 一种得知通信会话存在的方法，所述方法包括：

由第一用户发起通信会话，所述通信会话与第二用户的识别信息相关联；

检测由所述第二用户做出的异常动作；

根据对异常动作的检测，由所述第二用户针对与所述识别信息相关联的通信会话的存在进行轮询。

120. 根据权利要求 119 所述的方法，其中，所述异常动作是滚动事件。

121. 根据权利要求 119 所述的方法，其中，所述异常动作是在没有可识别功能的情况下选择网站的一部分。

122. 根据权利要求 121 所述的方法，其中，选择包括点击所述网站中没有链接的部分。

123. 根据权利要求 119 所述的方法，其中，异常动作包括使用多个快速捏合运动来触摸屏幕。

124. 根据权利要求 119 所述的方法，其中，所述通信会话是协同浏览器会话。
协同浏览与其他形式的信息共享的集成

[0001] 相关申请的交叉参考

[0002] 本申请要求于 2013 年 3 月 11 日提交的美国临时专利申请 No. 61/776,017 和 2012 年 7 月 19 日提交的美国临时专利申请 No. 61/673729 的优先权，上述申请的其全部内容通过引用并入本文。

[0003] 版权声明

[0004] 本专利文献的部分公开内容包含受版权保护的题材。版权所有者不反对任何人以本专利文献或专利公开呈交于专利商标局的文档或记录中的确切形式来进行静电复制，但将保留其他任何方面的一切版权。

技术领域

[0005] 本领域涉及通信会话，更具体地，涉及一种用于将协同浏览与其他形式的信息共享进行集成的方法和装置。

背景技术

[0006] 凭借网站销售和其他客户互动的公司往往为客户提供支持，由此，如果访问者在浏览网站时出现困难，那么他能够与例如客户服务代理的现实人进行交互。代理可使用基于文本的通信会话（聊天会话）或使用基于语音的通信会话（电话呼叫）与访问者进行交互。

[0007] 例如，由银行运营的网站包括那种相连于帮助服务的链接，以允许潜在客户与客户服务代表进行聊天或者交谈从而获取有关其帐户可用功能的更详细信息。当客户点击帮助按钮时，启动聊天会话或语音会话，这将促成客户代表与客户相连通。输入聊天会话中的消息在客户和客户代表之间继，或者代理和客户进行实时会话，这使得客户能够说明问题并获得代理的协助。

[0008] 当访问者在访问或使用网站的特定方面出现麻烦时，代理可尝试指导客户如何克服困难。不幸的是，代理所提供协助的能力在此情况下受到该代表无法看到客户所看的网站的阻碍。因此，从代理的角度来看，为代理提供能看到访问者所看到内容的能力将是有利的。

发明内容

[0009] 这里，提供了如下发明内容和在本申请末尾阐述的摘要，以介绍在下面具体实施方式中所讨论的一些概念。发明内容和摘要部分并不全面，并且不旨在勾勒下面提出的权利要求所阐述的受保护主题范围。

[0010] 协同浏览服务允许卖方的客户服务代理看到客户正在查看的网页。在确切知道客户所看到的内容的情况下，代理随后以更好的角度来提供有益的援助。通过在代理的视图中遮蔽客户的敏感数据来加强隐私，由此使得代理能够在不取得访问客户私人数据的情况下提供援助。在一个实施例中，诸如 JavaScript 的 ECMAScript 用于向代理镜像访问者的
浏览器状态。

[0011] 隐藏字段列表（以及潜在 div 和网页）用于防止访问者发送与意图包含敏感数据的字段相关联的信息，从而防止信息从访问者浏览器泄露。例如，可通过将 HTML 输入要素保存在代理端但访问者输入至该输入要素中的内容或状态不会传递给代理的方式，来对该 HTML 要素实施字段的隐藏。

[0012] 网页视图取决于（a）由 HTML 要素的层次结构代表的当前文档对象模型（DOM）以及（b）例如鼠标位置、滚动偏移量、暂停和焦点状态的状态信息。为了使代理端的视图与访问者端的视图同步，将页面的初始 DOM 树以及对其的任何后续移动都转发给代理，但同时移除 <script> 标签。这不必要求代理具有同步本地状态信息（例如，网络跟踪器 cookies、浏览器本地存储器）就能允许代理看到访问者所看的内容，但如果代理方浏览器要运行与访问者浏览器相同的脚本时就需要所述同步式本地状态信息。另外，对不影响文档对象模型（DOM）的变化（例如，与暂停状态有关的变化）分别进行检测并将其转发给代理，以使得代理能够模拟浏览器状态。

[0013] 通过代理所能获得的信息，可以多种形式建立协同浏览会话。示例性信息系统包括聊天软件、点击通话客户关系管理软件（其中访问者提供其电话号码以便代理进行电话回拨）以及其他软件系统。

[0014] 协同浏览会话可结合于其他形式的互动信息共享会话（包括屏幕共享会话），以使得代理在与访问者互动的同时能够利用额外的通信形式。还可以在传统电话或第三方聊天会话中通过口头交流或输入参与者之一点击网页上的对象时出现的代码的方式来建立会话。

附图说明

[0015] 本发明的各个方面将在所附权利要求书中具体指出。本发明由下面附图中的示例示出，其中相同要素以相同的附图标号表示。以下附图公开了本发明的各种实施例，这仅作说明性目的而不意在限制本发明的范围。为清楚起见，每个附图中并未对所有的部件进行标记。在附图中：

[0016] 图 1 和 2 是示出了协同浏览式客户支持系统的参与者之间的信息流的功能性框架图；

[0017] 图 3-17 显示了能将与访问者相关联的协同浏览会话连接至由代理定位的网站的方法；

[0018] 图 18 显示了协同浏览与社交媒体集成的示例；

[0019] 图 19-20 显示了由代理向访问者突出显示信息的示例性方法；

[0020] 图 21 是示出了用于确定增量更新的示例性过程的流程图；和

[0021] 图 22-28 显示了将屏幕共享会话与协同浏览进行集成的示例性方法。

具体实施方式

[0022] 详细说明

[0023] 下面的详细描述阐述了许多具体细节，以提供对本发明的一个或多个实施例的全面理解。然而，本领域的技术人员应当理解，本发明无需这些具体细节就可以实施。在其他
实例中，为了不将本发明造成混淆，没有对公知的方法、过程、部件、协议、算法和电路进行详细描述。

[0024] 定义

【0025】客户：签订了协同浏览服务的协同浏览服务客户。示例性客户为图1的卖方（例如，拥有卖方网站16的公司）。

【0026】网站：由协同浏览所支持的客户所拥有的网站。

【0027】团体ID：分配给每个将进行协同浏览的客户网站的唯一ID。

【0028】访者：导航至该网站的任何人。

【0029】代理：支持访者者在网站上开展各种任务（例如，完成购买）的代表人员。

【0030】遮蔽字段：在窗口中或网站上的应当从视图中遮蔽且对代理不可见的输入字段。例如，遮蔽字段可包含含有例如访者信用卡号码的敏感数据。

【0031】遮蔽对象：在访者的网络浏览上从代理的输入中遮蔽掉的输入字段。例如，遮蔽对象可包括字段、窗口、div、页面、链接、按钮等，以及需要保持在访问者控制之下的其他对象。不允许代理输入、修改或点击在远端控制中运行的遮蔽对象。

【0032】CServer：运行协同浏览会话的安全服务器，其接受来自访者者的会话更新并将其转发给代理。

【0033】协同浏览网络服务：对由访者者和代理访者的协同浏览会话进行控制的网络服务器。协同浏览网络服务可以与Cserver搭配或者可以是网络上的独立实体。

【0034】协同浏览概述

【0035】图1示出了协同浏览场景的网络图，其中访者者浏览器10的内容在代理浏览器12中是可见的。在图1所示的实施例中，访者者浏览器从卖方网站16获得页面14。箭头1示出从网站至访者者的页面下载。协同浏览服务18通过访者者浏览器向代理网页浏览器12（箭头3）中继网页更新（箭头2）来协助访者者浏览。为了使访者者浏览器10能提供这些更新，将脚本20（在一个实施例中是JavaScript）从卖方网站（箭头4）或从协同浏览服务（箭头5）下载至访者者浏览器。备选地，JavaScript可保留为访者者浏览器中的脚本。为了实现隐私，如下面将更详细讨论的那样，将遮蔽要素列表22下载到访者者浏览器中。类似于脚本那样，遮蔽要素列表可从卖方网站下载（箭头6），或者可从协同浏览服务下载（箭头7）。遮蔽要素列表向脚本指示出哪些要素（例如，DOM 的此类要素）不应在协同浏览会话上传输。代理浏览器也可从协同浏览服务下载（箭头8）页面24，其中访者者浏览器内容的视图将在协同浏览会话期间显示。

【0036】尽管在将要描述的实施例中，脚本20描述为是使用JavaScript来实施的，但是也可以使用其他形式的脚本语言。例如，JavaScript是ECMA-262规范中由ECMA国际制定的脚本语言标准的实施方式。该标准的其他实施形式包括JScript和ActionScript。因此，尽管将要描述的实施例中使用了JavaScript，但是本发明并不局限于这个特定实施形式，也可以使用其它形式的脚本。

【0037】协同浏览方案使用代理，能够使访者者当前出现的页面（卖方网站）。由于协同浏览会话仅取决于在网页浏览会话期间下载的JavaScript，因此不要求访者者手动安装任何软件或插件，或者解除任何安全警告对话框以便让代理看到访者者浏览器。同样，由于JavaScript是由卖方网站所提供的网页的一部分，因此JavaScript将不会延用到任何其
他网页，这意味着代理将不能看到来自其他网站的任何页面或访问者桌面上的任何其他页面。

【0038】 如下面更详细描述的那样，协同浏览方案使得代理能够准确看到访问者正在查看的内容。例如，如果该网页对访问者是不完全可见，则代理将仅能看到该网页可见的部分。这种视图将随着访问者滚动页面或调整页面大小而自动更新。在访问者打开多个选项卡的情况下，代理仅能够看到具有焦点的选项卡，以便代理看到访问者正在查看的内容。代理还可以看到访问者的鼠标指针的位置，这允许代理知道，在与网页交互的同时访问者关注哪里。如果代理的浏览器是代理的计算机上的活动窗口，那么该代理也能够看到访问者一端目前关注哪些输入字段。如果访问者打开选择框，那么该选择框也会在代理那端开放。

【0039】 为了使代理能够以可视方式与访问者进行交互，允许该代理在代理浏览器 12 上点击网页的特定要素或区域。当代理人这样执行时，相应的要素将同时在代理浏览器和访问者浏览器中突出可见，从而使得代理可将访问者的注意力引导至网页的特定部分。例如，突出可以采取用环绕特定要素绘制框的形式，或者是能在视觉上区分这些要素的一些其它方式。优选地，在代理按住其鼠标按钮的同时这种突出对访问者是可见的。释放后几秒钟内或者其它指定的短时期内，不需要代理的进一步行动，突出就会自动消失或消退。

【0040】 选地，代理能够提供对访问者浏览器 10 的输入。例如，代理能够将信息输入到访问者浏览器 10 上所显示的网页 14 的选定要素中，改变下拉列表的设置，选择单选按钮或复选框，或者与页面上的输入要素进行互动。取决于方式，代理能够代表客户来点击按钮或链接。

【0041】 为了保护访问者的敏感数据（例如，访问者的指定信用卡号和其它敏感数据），将包含例如信用卡号码的敏感数据的格式字段或网页区域从视图中掩盖掉，以便不向代理提供敏感数据。由于数据从未离开访问者浏览器，因此协同浏览服务将永远不会访问访问者敏感数据或与该数据进行交互。访问者端的掩盖数据相应地不需要访问者将敏感数据发送到网络上或者放弃对敏感数据的控制，从而使得该数据永远不会离开访问者的控制。

【0042】 选地，掩盖字段可被访问者识别以被访问者能够知道哪些字段被掩盖而哪些字段未被掩盖。提供这种性质的指示（例如以绿色框框出要掩盖的字段）可以为访者者提供安全感，使其知道这些字段中的信息对代理是不可见的并且不会离开访问者浏览器。

【0043】 由于以上描述的协同浏览服务基于 JavaScript 而不是例如 Flash、Java 或 Silverlight 的那种所有网络浏览器均不支持的平台，因此协同浏览服务能够为代理提供查看访问者浏览器的能力，即使是在访问者使用移动设备（例如，智能手机或平板电脑）来访问网站的情况下。同样，协同浏览方案也适用于 PC/Mac 苹果机。同一地，上传访问者 DOM 和状态信息（相比于仅发送访问者页面 URL 来说）使得在访问者访问卖方网站的同时能够在任意点开启协同浏览会话，而不需要访问者重新创建他 / 她的浏览器状态。最后，由于卖方网站的唯一变化是包含了脚本，因此协同浏览服务使代理能够在少数改变或不改变卖方网站的情况下为客户提供基于协同浏览的支持。

【0044】 实施方案

【0045】 根据实施例，协同浏览服务取决于经卖方同意而插入到卖方网站的脚本标签。当访问者加载卖方网页时，此 JavaScript（例如 cobrowse.js）可从网站传递至访问者，或者可直接从协同浏览服务传递至访问者。一旦协同浏览会话开启，此 JavaScript 会促使访
同者浏览器从访问者向协同浏览服务上传卖家网页 HTML 以及其他数据，相应地，协同浏览服务将这种数据表示为网页来供代理查看。

备选地，不使用 JavaScript（例如 cobrowse.js）从网站加载到访问者，而是手动地将 JavaScript 代码安装为浏览器插件。具有插件的优点是，不必要是特定的网站，并且相应地允许人员对尚未被修改为包括协同浏览脚本的网站进行协同浏览。具体来说，无论人员（“领导者”）访问的是什么网站，包含在插件中的 JavaScript 都将使浏览器状态转发给协同浏览服务。这原因在于领导者想要与其他的“追随者”分享他们的网络经验的情况下可能是有用的，因此所有人可以享受在视觉上跟随领导者进行网上（例如，与社交购物有关的网页）冲浪的社会经验。

在一些实例中，卖家网站可能已经具有其他现有的 JavaScript，例如来自聊天工具或点击通话功能的 JavaScript。在卖家的许可下，该现有的 JavaScript 可用于调用 cobrowse.js 脚本，这种做法的好处是，卖家可以在根本不对卖家网站进行修改的情况下试用或部署协同浏览方案。

在一个实施例中，如果协同浏览会话已经启动，那么：a）在协同浏览会话启动时刻，或 b）当从卖家网站的页面完成加载时，JavaScript（在本示例中为 cobrowse.js）执行以下内容：

1. 获取拖放元素的描述符。拖放元素的描述符可动态地从运行的服务器中（通过 AJAX 或的 WebSockets）来检索，或可作为 JavaScript 变量嵌入到 cobrowse.js 文件中；

2. 周期性地进行：

a. 克隆文件头要素，如果页面上还没有指定库标签，那么将表示任意相关的 URL 库的 <base> 标签插入到网页上，移除脚本标签记并将所克隆的层次结构呈现为 HTML。如果页面上已经指定了 <base> 标签，但是在它前面存在具有相关 URL 的 <link> 标签，那么将这些 <link>URL 转换为绝对 URL。备选地，添加 <base> 标签以及移除脚本标签可以在服务器端上由协同浏览服务来执行。将这些任务卸载到访问者浏览器上也可以使协同浏览会话服务器处理更多的会话。

b. 克隆文档 body 要素，移除脚本标签和隐藏输入，以虚拟数据替换拖放元素的值，并将克隆的层次结构呈现为 HTML。许多页包含一个或多个带有会话状态信息的隐藏输入，所述会话状态信息不会在视觉上影响页面。如果访问者端上绘制有气泡显示（gesture），那么由于在代理侧上再描绘姿态显示是多余的，因此 cobrowse.js 也会将姿态要素移除。备选地，脚本标签的移除可以在服务器端上由协同浏览服务来执行。将这些任务卸载到访问者浏览器也可以使协同浏览会话服务器处理更多的会话。

c. 遍历所有的用户输入对象（文本字段、单选按钮、复选框、下拉列表），并收集用户的任何数据输入。还可以收集画布对象的当前状态。

d. 检测访问者鼠标位置。

e. 检测访问者滚球偏移量。

f. 检测访问者浏览器的尺寸。

g. 检测页面上的哪个要素具有焦点。
h. 检测是否有，任何选择框打开。

i. 将上述所有内容（如果改变）发送至协同浏览服务。数据可经由 XMLHttpRequest 或 XDomainRequest (AJAX)、WebSockets 或者跨域通信的一些其他手段来发送。这个步骤中的数据发送可以在传输之前进行编码和 / 或压缩。任选地，如果仅数据的一部分发生了变化，则可以仅发送变化部分。

注意，(a) 至 (h) 是周期性重复的，这是因为运行在访问者浏览器中的 JavaScript 可对 DOM 进行修改。

在上述 (a) 和 (b) 中将脚本从上传至协同浏览服务的 HTML 中移除，从而使得知脚本下载给代理时其不运行。如果脚本修改了客户端浏览器中的文件要素层次结构，那么层次结构的变化将经由协同浏览服务反映给代理，以使得脚本不需要直接在代理浏览器中运行。

Cobrowse.js 可由卖方或协同浏览服务主导。这由图 1 中的箭头 4 和 5 来表示。同样，由访问者获得的掩蔽要素列表可从卖方网站 (箭头 6) 或从协同浏览服务 (箭头 7) 中检索。由于访问者获得的掩蔽要素列表针对网站是特定的，因此当由网站 / 为网站创建掩蔽要素列表，以指定出网站样式表中应不被传送到协同浏览会话的要素列表。在一个实施例中，掩蔽要素列表包括网站的层叠样式表 (CSS) 的选择器列表。该列表被提供给访问者以保护访问者的隐私信息。在一个实施例中，为每个网站创建列表，但是也可以针对那种协同浏览器服务性将会在上面实现协同浏览会话的每一网站页面来制作唯一的列表。

可通过允许管理员或其他人在视觉上识别卖方网站的要素并点击管理用户界面（例如，工具提示）来指定该可视要素是否应当被屏蔽以保护敏感信息或者不需要被屏蔽，来实施对待掩蔽要素的选择。一旦管理员已经选择或屏蔽的要素，则基于该输入创建 MaskedElements 描述符。管理用户界面可在 cobrowse.js 中实现，或可由卖方提供的软件来实现，以使卖方能够选择网站中要屏蔽的与协同浏览会话有关的字段。为了保护此列表，卖方需要在对卖方 MaskedElements 描述符进行修改之前进行登录或以其他方式进行验证。

尽管使用掩蔽要素列表能够无需修改网站即对一组要素进行掩蔽，但也可以创建掩蔽字段定义，使它自动促使那些具有任意定义的属性的对象被隐藏。例如，给定了掩蔽字段定义 *[co-browse hide = true]，然后，例如 co-browse_hide = “true” 的属性可用于识别网站上的应当被隐藏的要素。

差异更新

通常，运行在网络页面上的 JavaScript 响应于各种事件（例如，页面加载时，定时器或用户操作）会修改页面的 DOM。典型的示例是旋转图像 marquis (其中以规则时间间隔改变所显示图像)，或者弹出菜单 (其通常隐藏，但响应于用户点击而变为可见)。

为了提高效率，访问者保存整个 DOM 框的快照，并寻找差异。在可能的情况下，访问者 JavaScript 向页面发送关于增量改动的信息，以避免重新上传整个页面 HTML。当代理第一次查看该页面时，代理 JavaScript 从 CServer 下载初始的页面 HTML，然后应用从初始上传页面开始的任何增量更新。一旦代理正在查看网页，那么随着经由 Cserver 从访问者接收新的增量，任何新的增量变动都被立即推送给该代理。

通常，相同要素会在访问者端上反复变化。如果 Cserver 要存储所有这些由访问者报告的变化的话，那么当代理第一次连接到会话时该代理将接收到涉及屏幕的相同部位
可能导致那个屏幕部分的闪烁的一套更新。同样地，如果代理检索访问者所看到到的新页面的时刻与代理页面连接到 Cserver 以检索对页面的后续更新的时刻之间存在延迟，那么在由于后续更新施加至代理视图而导致屏幕部分闪烁的同时，可将访问者屏幕的多个变化转发至代理。

[0071] 因此，在实施例中，访问者累积所有的差异，并转发自完整页面已发送开始的累积差异。为了产生累积差异集合，访问者的 JavaScript 丢弃已由 DOM 的其他改变所覆写的无关的变化。

[0072] 差异更新的类型

[0073] 目前，存在三类由访问者检测到的增量差异：属性差异 (ATTR diff)、内容差异 (CONTENT diff) 和 HTML 差异 (HTML diff)。

[0074] 属性差异

[0075] 属性差异表明要素的属性集合发生变化。可以添加、修改或删除属性。ATTRdiff 指定新的属性集合。任选地，ATTRdiff 可以更精细粒度来指定对那些属性进行添加、修改或删除。

[0076] ATTRdiff 的 JSON 表示的示例为：

```
[0077] {
[0078]  "diff":"ATTR",
[0079]  "path": "#account>div:nth-of-type(1)>table:nth-of-type(1)>tbody:nth-of-type(1)>tr:nth-of-type(1)>td:nth-of-type(1)>p:nth-of-type(1)",
[0080]  "data": { "style": "color:red", "class": "boldparagraph" }
[0081] }
```

[0082] 在本示例中，“diff”表示增量差异的类型，“path”表示通向属性已改变的要素的 CSS 路径，而 “data”表示要素的当前属性集合。为了处理 ATTR diff，代理的 JavaScript 移除了要素的所有现存属性并采用 ATTR diff 中所指定的所有属性。

[0083] 内容差异

[0084] 内容差异表明要素的文本内容发生变化。要素必须仅含有文本节点而不包含要素。CONTENT diff 的 JSON 表示的示例为：

```
[0085] {
[0086]  "diff":"CONTENT",
[0087]  "path": "#account>div:nth-of-type(1)>table:nth-of-type(1)>tbody:nth-of-type(1)>tr:nth-of-type(1)>td:nth-of-type(1)>p:nth-of-type(1)",
[0088]  "data": "somewhere over the rainbow"
[0089] }
```

[0090] CONTENT 差异的示例中，“data”表示要素的新文本内容。为了处理 CONTENT diff，代理的 JavaScript 将要素现存的文本内容替换为 CONTENT diff 中所指定的当前文本内容。

[0091] HTML 差异
如果数目、类型或子节点的值发生变化则适用第三类差异，即 HTML diff。HTML diff 的 JSON 表示的示例为：

```json
{
  "diff": "HTML",
  "path": 
    
  "data": "abc<def>ghi"
}
```

其中，“data”是由 #anid 识别要素内的当前 HTML 标记。目前，当检测到 HTML diff 时，访问者的 JavaScript 会发送新的完全更新。备选地，一些 HTML diff 可作为差异更新来被处理。

累积差异
随着 DOM 发生变化，访问者向 Cserver 发送这些变化的 JSON 表示。Cserver 将增量变化转发给已接收到访问者页面并与 Cserver 具有开放式 WebSocket 连接的任何代理。为了确保尚未检索到访问者页面或尚未与 Cserver 建立 WebSocket 连接的任何代理都能得到整组变化，Cserver 必须存储自访问者首次发送页面起出现的累积差异集合。通常情况下，如在 marquis 的示例中，网页 DOM 进行多次改变。在发生这种情况时，Cserver 可通过要应用的增量的任意长列表来终止。代理加载会话将需要下载并采用这个长列表。这可能表现欠佳并可能导致代理视图的闪烁。相同的页面要素通常每次以相同的方式来变化（例如，<img> 标签的“src”属性会在一系列图像中旋转），这导致针对同一要素的一序列 ATTR diff。为防止增量 DOM 变化的积累，访问者的 JavaScript 不仅会发送最新的增量变化，而且还会发送合并的累积变化（累积差异）的集合。如果相同的要素具有适用于它的相同的改变类型，那么最新变化将覆盖累积差异中的旧变化。

例如，

```json
{
  "diff": "ATTR",
  "path": 
    
  "data": {
    "style": "color:red"
  }
}
```

后面是:

```json
{
  "diff": "ATTR",
  "path": 
    
  "data": {
    "style": "color:blue"
  }
}
```

将合并到单个累积差异中:

```json
{
  "diff": "ATTR",
  "path": 
    
  "data": {
    "style": "color:blue"
  }
}
[0120]  
[0121] 在将累积差异收集并合并到访问者端的实施例中，当代理查看网页时，查看器将检索并显示原始的页面 HTML。然后，查看器通过 WebSocket 连接来检索当前的累积差异集合，然后将这些差异应用于视图。随着增量的差异从访问者端到来，通过 WebSocket 连接来接收后续的所有增量差异。

[0122] 另一种实施方案是将增量差异合并到服务器端。这将不需要访问者上传具有每次 DOM 变化的全套累积差异，但却增大了 CServer 的负载量。

[0123] DOM 树差异算法

[0124] 访问者的 JavaScript 存储了每次向 Cserver 发送更新的 DOM 树的副本。当发送新的更新时，访问者的 JavaScript 将当前 DOM 树与发送的具有前一更新的 DOM 树进行比较，计算 ATTR diff、CONTENT diff 和 HTML diff 的集合。图 21 显示了比较算法 Diff(elem1, elem2)。注意，图 21 所示的 Diff() 算法是递归方法的，其初始由分别作为 elem1 和 elem2 的先前 DOM 树和当前 DOM 树的文件要素根调用。

[0125] 在 HTML 文件中，HTML 要素可以既包含要素又包含文本子节点。要素是嵌套式 HTML 要素，其相应可以包含其他节点或要素，而文本子节点则包含文本跨距。如果要素不包含其他要素（并仅有一个或零个文本节点）的话，那么文本节点中的文本变化可以由 CONTENT diff 来表示。如果要素即包含要素又包含文本节点的话，那么任何文本节点中的文本变化都将由母要素的 HTML 变化来表示。

[0126] 如图 21 所示，比较算法 Diff(elem1, elem2) (2100) 开始于通过检查来确定是否存在 ATTR diff, 从而确定要素 1 的属性是否与要素 2 的属性不相匹配 (2102)。如果属性上存在差异，那么该算法用要素 2 的属性创建 DIFF diff。

[0127] 该算法下一步比较子要素的数量和类型 (2104)。如果是子要素不同，则算法将用子要素 2 的内部 HTML 来添加 HTML diff (2106)。此时，算法已经检测到了 HTML 差异并相应停止递归 (2108)，并且转发更新后的 HTML。

[0128] 如果子要素的数量和类型被确定为相同 (2104)，则该算法将确定是否存在零个子要素 (2110)。如果不存在子要素，则该节点为纯文本节点。该算法因此将对要素 1 的文本内容和要素 2 的文本内容 (2112) 进行比较。如果这两个要素的文本相同，则这两个要素之间没有差异，算法将停止递归 (2114)。如果这两个要素之间存在差异，则算法将用子要素 2 的文本内容创建 CONTENT diff (2116)。

[0129] 如果存在多于零的子要素 (2110)，则该算法将比较每个子节点的数量、类型以及节点值 (nodeValue) (2118)。如果这些值不同，那么算法将用子要素 2 的内部 HTML 添加 HTML diff (2106)。此时，算法已经检测到了 HTML 差异并相应停止递归 (2108)，并且转发更新后的 HTML。

[0130] 如果算法未检测到 HTML 差异 (2118)，那么它针对要素 1 和要素 2 的每一对相应子要素进行递归 (2120)，以寻找 DOM 树中更底级要素的变化。通过这种方式，对整个 DOM 树进行递归以寻找 DOM 的增量更新，该增量更新可用来创建用于转发至协同浏览会话的更新。

[0131] iFrame 要素

[0132] 访问者的网页可能包含引用了客户端网站上的另一页面的 iframe 要素（框架要素）。当访问者的 Cobrowse.js 将页面呈现为 HTML 时，iframe 要素将呈现为 <iframe src
说明书记

iframe 页面的内容不包括在内。然而，Cobrowse.js 的单独副本却如同客户端网站上的所有其他协同浏览网页那样地包含在 iframe 源页面中。当 iframe 页面进行加载时，iframe 中的 Cobrowse.js 将 iframe 页面数据作为会话的“子会话”而上传给协同浏览服务。iframe 中的鼠标的位置、姿态、输入、滚球位置、鼠标点击等都如同主会话那样传送至 Cserver。

[0133] 当上传网页 HTML 时，访问者用 iframe 子会话的 URL 来替代 iframe src 属性。在代理端 iframe 将检索子会话视图。代理 iframe 视图也将包含代理查看器 JavaScript(agent viewer JavaScript)CobrowseViewer.js 的额外拷贝，这将基于来自 Cserver 的更新而持续地更新视图。

[0134] 代理视图

[0135] 代理此时使用标准网络浏览器或者可能使用以插件或一些其他手段强化的网络浏览器（用于提高性能）来登录到协同浏览服务中。代理浏览器请求客户页面的代理视图。提供给代理的页面包含与客户浏览器（如图 1 所示）具有相同尺寸的 iframe iframe src URL 通过<script> 标签引用嵌入在其中的“cobrowseviewer.js”，来检索由协同浏览服务使用的具有客户发送的头部 HTML 和主体 HTML 的页面。如果访问者滚动了页面，那么协同浏览服务会将<div> 插入到已经相对于页面左上角偏移了滚球偏移量的页面中。div 具有标识符“glance_scroll”并且 iframe src URL 包括相对部分“#glance_scroll”，由此使得当页面加载到框架中时页面会立即偏移，以使得<div> 位于框架的左上角。这种做法使得无需等待网页上的 JavaScript 来滚动代理视图，代理视图就能立即定位在正确的滚球偏移量处。

[0136] Cobrowseviewer.js 运行在代理浏览器上，并且定期地从协同浏览服务检索增量 DOM 变化、滚动偏移量、鼠标位置、浏览器尺寸、元素状态以及输入字段值。如果存在可用于网页 HTML 的完整更新，那么协同浏览服务还将通知代理。Cobrowseviewer.js 执行以下操作：

[0137] 1. 如果存在用于网页 HTML 的完整更新，则 Cobrowseviewer.js 刷新整个框架以利用当前 HTML 获得新的视图。（备选地，该页面可使用更新后的头部 HTML 和主体 HTML 自己进行重建）

[0138] 2. 否则，cobrowseviewer.js：

[0139] a. 将任何的增量变化应用到 DOM；

[0140] b. 将窗口（该窗口在 iframe 内）滚动至与客户的浏览器窗口相同的偏移处；

[0141] c. 在与客户的鼠标光标相同的坐标处显示图像以代表客户鼠标；和

[0142] d. 将输入字段的值设置为指定值；

[0143] e. 如果代理浏览器的窗口是桌面上的活动窗口，则将焦点设置到访问者的焦点要素；

[0144] f. 如果访问者打开了选择框，则在代理视图中打开相同的选择框。

[0145] 姿态示意

[0146] 在一个实施例中，代理所采取的行动被 cobrowseviewer.js 捕获并传送给协同浏览服务。相应地，协同浏览服务向访问者中继所捕捉的输入，以突出网站的方面或者采取其他行动与显示在对访问者浏览器 10 上的网站 14 相连。
在每一级产生 CSS 选择器部分的算法为:

获取要素的 ID，如果具有的话，并通过转义特定字符或将其转换为 Unicode 转义序列来将其所对应的 ID 转换为有效的 CSS ID。

如果 ID 唯一地识别其直接父系内的要素，那么这一级别的 CSS 选择器部分为 #theid。

如果 ID 唯一地识别整个文件中的要素，则退出。该要素的 ID 作为路径的唯一起点。

如果要素是文件主体，那么 CSS 选择器部分则就简单地是 “body”。退出；该主体是路径的起点。
否则，确定该要素在其相同类型的同族要素列表中的索引位置n。这一等效的CSS选择器部分则是标签nth-of-type(n)

下面提供唯一的CSS选择器的几个示例:

* #anegative
* body>table:nth-of-type(1)>tbody:nth-of-type(1)>tr:nth-of-type(1)>td:nth-of-type(2)
* #content>div:nth-of-type(1)>form:nth-of-type(1)>table:nth-of-type(1)>tbody:nth-of-type(1)

代理还可（经过或不经过访者者的允许）进行远程控制。在这种模式下, 代理的指针位置（与访者的指针不同）可显示给访者者。代理也可以向字段中输入值, 另向访者浏览器栏的字段中输入数据, 并且还可以点击链接、按钮等。Cobrowseviewer.js捕捉按键敲击并将输入传送给访者浏览器。访者浏览器捕获相关字段中的输入, 以使代理能够将数据输入到该网站的字段中。通过在协同浏览会话期间让代理能够控制访者浏览器, 使得代理能够通过导航网站来为客户提供直接援助。

为了防止访者在访者浏览器中采取特定动作, 在一个实施例中, 针对该网站定义了能防止代理向指定的表格、div、页面等输入数据的掩蔽对象列表。同样, 掩蔽对象列表将阻止代理点击某些链接或按钮, 例如基于点击通过或许可协议的“我同意”按钮。此外, 代理被限制在特定的页面, 并且不能从其他网站来远程控制网页。

反向协同浏览

在另一种场景中, 代理可发起与访者的协同浏览会话。在这种情况下, 代理可打开浏览器, 导航至网页的特定部分, 然后启动与访者的协同浏览会话。例如, 访者者在网页上出现麻烦, 访者者联系客户支持; 代理并非要告诉访者者去哪里，而是定位到正确的网站, 并启动了协同浏览会话（使用cobrowse.js）。协同浏览查看器JavaScript（即co-browseviewer.js）通过聊天而传送到访者者, 从而促使访者者端的用于将访者者连接至网站的选项卡或新的浏览器打开。然后, 代理可驱动该浏览器或允许访者者来控制该浏览器。

**悬停状态**

如果访者者允许其鼠标指针在卖方网页的要素上方保留给定的时间段, 那么浏览器可将这个动作解释为“悬停”。网页可设计为根据是否处于“悬停”状态来将不同类型的属性应用于要素。某些形式的悬停造成DOM的变化, 这将反映给代理。例如, JavaScript“onmouseover”可用于指定根据悬停所应当实现的效果类型。当“onmouseover”事件发生时（例如，显示子菜单），这将导致DOM的变化, 并且正常的协同浏览DOM更新将会将该变化反映给代理。然而, 其他形式的悬停则不会导致DOM变化。例如, CSS样式可定义为, 当访者者的指针悬停在要素上时, 将显示与要素相关的选项菜单。这允许在网页中动态地提供附加信息, 而不需要访者者点击链接, 从而在页面之间移动。

与此相关的问题是, 悬停状态不会影响DOM, 因此简单地通过保持代理DOM与访者者同步, 就能使得视觉上的变化不会反映在代理端上。由于无法对代理的实际鼠标指针进行定位以匹配访者者的鼠标位置, 因此相同的悬停状态不会发生在代理端上, 并且代理将不会看到基于访者者的悬停状态所显示出的网站各方面。如果访者者端的脚本修改了访者
者浏览器中的文件要素层次结构，那么层次结构的变化将通过如上所述的同源浏览服务反映给代理。然而，悬停状态不会改变层次结构，并因此对代理是不可见的。

[0169] 因此，实例中的情况是，通过代理无法看到的悬停状态的事件，可以向访问者提供网站的特征。当代理正试图通过网站来帮助访问者时，这种无法看到访问者所看内容的做法会产生问题。

[0170] 根据实施例，为卖方网站提供支持的代理配备有以 CSS 选择器形式的针对网站的可悬停要素列表。该可悬停要素列表可被定义一次，并且可供为卖方网站提供支持的所有代理使用。类似于定义掩盖要素列表的方法，可以通过多种方式来创建可悬停要素列表。在协同浏览网站中一起检索可悬停要素列表与掩盖字段定义。为在代理端复制悬停状态，对访问者的鼠标位置进行监控并将该位置用于选择性地重新创建代理端的悬停状态。下面对一种实现方法进行更详细的描述。

[0171] 使用样式表单来定义网站的外观。样式表单定义了应用于要素和要素类别的样式。一个可定义的方面是对要素/类别的悬停状态进行定义，即，当访问者将鼠标悬停在要素上方时会发生什么。例如，网站可例如通过下表 1 中所列举的如下代码来指定当用户将鼠标悬停在一级菜单选项卡上时会出现二级菜单：

[0172] 表 I

```css
.navbar .level1item:hover .level2menu {
    display:block;
}
```

[0173] 代理的主机实际鼠标位置不会跟随访问者鼠标。因此，代理浏览器不会检测到“悬停”状态，由此代理端上不会显示二级菜单。有几种方法可以为代理解决这个问题。例如，针对光标自身下方的要素，访问者的 JavaScript 可以调用 getComputedStyle() 并将其发送给代理的浏览器，由此让代理将这种计算方式应用到代理端的同源要素上。但是，除非是不仅针对光标下的要素而且还针对所包含的所有要素来调用 getComputedStyle，否则在嵌套式悬停要素的情况下这将不起作用。

[0175] 需要使客户端网站所依赖的两个配置片段就位; 即，悬停样式：1) “可悬停”要素列表，和 2) 仅应用于代理视图的代理 CSS 样式集合。例如，可创建工具来扫描网站的样式表单并产生在网站上可悬停的要素列表。

[0176] 可悬停要素是那种当鼠标悬停在它上方而具有适用于它（或者它的同族或子族之一）的样式的要素。可悬停要素通常可由任何有效的 CSS 选择器来指定标签名和 ID 或着类别。可悬停要素可基于每个用户来定义。表 II 显示了如何为可悬停要素指定网站样式定义的示例：

[0177] 表 II

[0178]
例如，以下样式定义：

```css
#div1:hover
{
  /*每当鼠标悬停在由div1定义的元素上时，这种样式定义就适用于任何具有ID“div1”的元素*/
}
#div2:hover .c2
{
  /*每当鼠标悬停在由div2定义的元素上时，这种样式定义就适用于具有CSS类c2的元素，其中这种CSS类c2是具有ID div2的元素的后代*/
}
.c1:hover
{
  /*每当鼠标悬停在具有CSS类c1的元素上时，这种样式定义适用于任何此类元素*/
}
h1.c3:hover p
{
  /*仅当鼠标悬停在具有CSS样式c3的<h1>元素上时，这种样式定义才适用于任何作为h1元素的后代的<p>元素*/
}
```

[0179] 在本示例中，“可悬停”将是:

- #div1
- #div2
- .c1
- h1.c3

[0184] 下表 III 中示出了那些包括在代理样式表单中的对应代理样式:

<table>
<thead>
<tr>
<th>#</th>
<th>代理样式表单中的对应代理样式</th>
</tr>
</thead>
<tbody>
<tr>
<td>.div1:hover</td>
<td>.div1_hover</td>
</tr>
<tr>
<td>.div2:hover..</td>
<td>.div2_hover .c2</td>
</tr>
<tr>
<td>.c1:hover</td>
<td>.c1_hover</td>
</tr>
<tr>
<td>h1.c3:hover</td>
<td>h1.c3_hover p</td>
</tr>
</tbody>
</table>

[0187] 为了使悬停状态复制到代理端，访问者端处的 JavaScript 观察访问者端的鼠标
位置并查看鼠标位置是否定位于任何“可悬停”要素上方。如果是，则代理需要对代理视图中的对应要素应用一种（或多种）悬停样式。访问者的 JavaScript 发送给代理的应用的样式列表。此列表中的每个条目包括：

[0188]  csspath: 通向已被悬停的要素的 CSS 路径。CSS 路径是使用上述算法来指定的。
[0189]  classnamn: 应用于被悬停要素的样式名称。样式名称是基于用于识别要素的 CSS 选择器来生成的。在上面的示例中，如果对要素匹配选择器 #div1 进行悬停，则样式名称将是“div1_hover”。如果对要素匹配选择器 .cl 进行悬停，则样式名称将为 cl_hover。对于一些网站而言，这种技术可能不够，可能需要扩展这种命名惯例或者允许管理员来指定应当与特定选择器相关的样式名称。

[0190]  代理循环访问这个列表，找到被 csspath 所识别的各个要素，并且对其应用指定的 classnamn，产生了与应用于访问者端上的悬停状态样式相同的视觉效果。

[0191]  当访问者悬停在要素时，该动作被检测到，并将命令转发给代理以通知代理悬停状态已被触发。通过向访问者传递可悬停“CSS 选择器”列表，以及让访问者浏览器确定所悬停的事项是否与选择器匹配，可以让 JavaScript 向代理传递指令，以使代理能够在代理端重新创建悬停状态。

[0192]  协同浏览服务

[0193]  协同浏览服务允许一个或多个客户服务代理实时地查看访问者对网站的网页浏览活动。还可以开发协同浏览服务的其它用途，而且协同浏览服务可以使用在人员想要实时地查看访问者对特定网页的网页浏览活动的情况中。因此，尽管所提供的示例是代理能够看到访问者所看的内容，但是本发明不局限于此特定用途。

[0194]  在本文中所描述的实施例中，除了例如信用卡号码或密码的指定拖拽字段的内容不能看到外，代理在其浏览器中确切地看到了访问者所看的内容。同样，由其他网站提供的动态内容如广告，其为来自卖家网站以外的其它网站上的内容，这对代理是可见的，但代理可能不会获得与访问者相同的广告。

[0195]  在一个实施例中，协同浏览服务实现了能确保安全性并提高协同浏览会话的集成度的几种机制。协同浏览网站具有例如登录和账户管理系统的到位的安全措施，这使得网站本身能对抗攻击。

[0196]  体系结构概览

[0197]  图 2 显示了示例性协同浏览服务的示例性体系结构概览。协同浏览服务依赖于嵌入在客户端网站中的<script>标签，该标签引用了宿主于协同浏览网络服务器的 JavaScript 文件。就该文件的目的来说，访问者端的这个 JavaScript 文件将称为 Cobrowse.js。也可以对 JavaScript 使用其他名称。

[0198]  当协同浏览会话开始时，Cobrowse.js 向协同网络服务器发送 StartCobrowse 请求（图 2 中的箭头 1）。Cobrowse.js 从访问者 100 向协同浏览网络服务器 110 传递客户端 ID，并接收唯一的会话 ID、Cserver 任务、拖拽字段及可悬停选择器列表。

[0199]  在一个实施例中，会话 ID 包括三部分：1) 客户端 ID，2) 会话密钥，和 3) 用于唯一性的随机数。还可以根据实施例来传递其他参数。

[0200]  Cobrowse.js 在浏览器会话 cookie 中存储会话 ID 和拖拽字段集合，并开始将会话数据上传到指定 CSserver 120 中（图 2 中的箭头 2）。CServer 相应地向协同浏览网络服务
器传递消息（图2中的箭头3），从而在数据库中记录下会话已开始的事实。
[0201] 要求将团体ID和会话密钥都结合到会话中。一旦登录，可根据代理的协同浏览组成员来确定团体ID。例如，代理组提供给一个或多个客户端的支持。当访问者位于客户端时下载到访问者端的JavaScript包含当会话开启时由客户上传至协同浏览网络服务器（箭头1）的客户团体。这使得协同浏览网络服务器能够得知哪些客户与特定的协同浏览会话相关联。在代理为多个客户提供支持的情况下，可代理可向协同浏览器提供客户团体ID，以选择与所支持客户中的一个客户相关联的协同浏览会话（图2中的箭头4）。
[0202] 针对会话密钥，有几种方式可将其指定在代理端。会话密钥可以是随机分配的字符串，或者可能是与访问者相关联的一些信息片段，例如用户ID或跟踪CookieID。代理可手动输入会话密钥，或者自动从CRM记录器的数据库中为代理提取该会话密钥。下面阐述示例性连接方案的附加细节。
[0203] 无论通过哪种方式，结果都是，代理将浏览器窗口打开至与协同浏览会话相关联的URL，例如：
[0204] https://www.cobrowse.net/cobrowse/AgentView.aspx？SessionKey＝ssnkey
如果没有登录，则会提示代理进行登录，然后重新引导回到代理视图。
[0205] AgentView.aspx通过代理团体ID和会话密钥在数据库中查找会话，并将代理重新引导至能产生会话中的代理视图的合适的Cserver上。代理视图包括嵌入式JavaScript文档，即CobrowseViewer.js，该文档经由HTTP和WebSockets与CServer通信（图2中的箭头5）。
[0206] CServer仅向代理浏览器发送用于浏览会话的HTML标记，被会话HTML引用的资源（例如，图像或样式表单）由代理直接从客户（或第三方）网站上下载。
[0207] 安全事项
[0208] 协同浏览服务收集并传送潜在的敏感访问者浏览信息，并因此将安全性设计为最高优先级。协同浏览架构处理以下安全事项：
[0209] 安全代理视图
[0210] 当代理视图加入会话时，网络服务器基于代理协同浏览团体来查找会话以及所提供的会话密钥。如果找到匹配的会话，则网络服务器返回会话ID，该会话ID中签署了保密的“服务器密钥”。服务器密钥是仅为CServer和网络服务器知道的256位的数字。CServer每60秒产生新的服务器密钥，并与网络服务器安全共享。当代理被重新引导至CServer以查看会话时，CServer会验证会话ID上的签名，如果签名有效则仅允许代理加入。这种方法保证了仅在a）代理得到网络服务器验证和b）代理是该会话所关联的团队中的成员时，查看会话的请求才会被允许。
[0211] 一旦代理参与了会话，安全（标示为仅用于HTTPS）会话cookie用于保持代理的会话。代理会话cookie包含用仅CServer知道的密钥来签署的会话ID。CServer密钥是通过协议中生成的，并且不会在任何地方保留。
[0212] 当WebSocket连接首次建立时，该WebSocket连接从代理端传递代理会话cookie值。如果请求中包括有效的代理会话cookie值，那么CServer仅接受与给定会话的WebSocket连接。
[0213] 在请求中不附带有效的代理会话cookie的情况下，CServer不会提供浏览会话数
说明书

据。这保证了只有以指定公司注册的协同浏览提供商账户的那些代理才能查看由那家公司网站的访问者开启的会话。

[0214] 安全数据传输

[0215] 使用 HTTPS 或安全 WebSockets 进行所有通信的做法确保了将所有数据安全地传输到身份已被认证的服务器中。这包括:

[0216] • 从协同浏览网络服务器将协同浏览 JavaScript 下载至访问者浏览器
[0217] • cdbrowse.js 和协同浏览网络服务器之间的通信
[0218] • cdbrowse.js 和 CServer 之间的通信
[0219] • CServer 和协同浏览网络服务器之间的通信
[0220] • 将会话数据和代理的 JavaScript 下载至代理浏览器
[0221] • AgentView.js 和 CServer 之间的通信（如果访问者网页是安全的）
[0222] • 从客户端将访问者会话资源下载至代理浏览器（如果访问者网页是安全的）
[0223] JavaScript 一体化

[0224] 使用来自协同浏览网络服务器的 HTTPS, 从协同浏览网络服务器将访问者 JavaScript 下载到客户端（即，客户向访问者提供 JavaScript 的位置）。HTTPS 确保了脚本本身来自于合法来源。客户端的责任是，确保协同浏览的 JavaScript 通过 HTTPS 嵌入到客户端中，甚至是非安全网页。为了鼓励这一点，协同浏览网络服务器将不会通过 HTTP 来提供访问者端 JavaScript，由此在测试过程中应当检测到任何通过 HTTP 访问 JavaScript 的企图。

[0225] 掩蔽字段

[0226] 由于掩蔽字段存储在协同浏览网络服务器的安全数据库中并且要经由与协同浏览网络服务器相连的安全连接而被检索，因此掩蔽字段定义不能被篡改。备选地，掩蔽字段定义可以如上述地由客户端提供，并由客户端来为访问者提供。掩蔽字段的内容从未泄露到访问者的浏览器，由此使得这种数据无法在传输过程中被拦截，甚至无法由具有合法会话 cookie 的代理从 CServer 来访问。

[0227] 掩蔽对象

[0228] 由于掩蔽对象存储在协同浏览网络服务器的安全数据库中并且要经由与协同浏览网络服务器相连的安全连接而被检索，因此掩蔽对象定义不能被篡改。备选地，掩蔽对象定义可以如上述地由客户端提供，并由客户端来为访问者提供。代理可阻止对掩蔽对象提供输入，从而防止代理对包括在掩蔽对象列表中的访问者浏览器的各方面进行控制。备选地，可以使用可控对象列表来指定访问者浏览器中可控制的那些对象，从而默认地将那些不包括在可控对象列表中的对象自动视为掩蔽对象，且不能从代理接收输入。

[0229] 安全的 CServer

[0230] 每个协同浏览服务器 120 是运行其自身的本地防火墙的独立 Unix 服务器。防火墙配置为仅允许从安全端口 443 上传入的通信量。CServer 上的任何文档或数据库都不保留会话数据。尽管图 2 示例了独立于协同浏览服务器 120 而实现的协同浏览网络服务器，但是其他实现方式可以将协同浏览网络服务器 110 的功能性方向与协同浏览服务器 120 相结合。

[0231] 无针对代理限制
需要注意，只要两个代理均具有同一团体的账户，就没有具备保护来防止一个代理查看本来意图针对另一代理的会话。即使当随机会话密钥是通过电话在访问者与代理之间私下交换的，也不会保护该密钥不被窥探，这是因为当访问者导航至客户端地址中的不安全页面时，该密钥会存储在以非加密方式上传的不安全 cookie 中。在另一实施例中，可以实现能使代理查看会话资源的其他限制。

针对采用内容分发网络的网站的代理 HTTP

如果代理会话严格使用 HTTPS 来访问访问者会话数据，那么代理浏览器将使用 HTTPS 来请求例如图像和样式表单的所资源。有些网站实施方案，特别是使用内容分发网络的网站，不能通过 HTTPS 来提供所有的资源。

为了解决这个问题，协同浏览服务可任选地配置为允许代理协议遵循访问者协议，以便当访问者导航至 HTTP 页面时，代理也是如此，反之亦然。代理端上保持有两个单独的会话 cookie，即，用于查看 HTTPS 页面的安全 cookie 以及用于查看 HTTPS 页面的非安全 cookie。尽管非安全 cookie 提供了最低的安全水平，但是代理会话不会免于通信量探查。备选地，代理服务可用于让代理能够查看此种情况下访问者正在观看的内容。

协同时联系点及方案

有许多方法能将代理连接到正确的协同浏览会话上。给定网站可能在任何一个时间点上都拥有成百上千的访问者。如果协同浏览 JavaScript 的下载与加载网页相关，那么针对每个访问者都将存在协同浏览会话。与特定访问者一起工作的给定代理需要选择正确的协同浏览会话，以使得该代理能够随着其特定客户的阅历而看到网站的状态。

因此，有必要让代理能够识别访问者并能正确地识别哪个协同浏览会话与所识别的访问者相关联。可供代理这么执行的特定机制将取决于特定的网站实现方案，以及代理正在使用哪些其他类型软件（例如，客户关系管理软件）。

代理可以使用由第三方或任选地由协同浏览服务提供的聊天服务与客户进行交互。同样地，代理可以具有运行在其实工作站在的客户关系管理软件，以供该代理对客户及其他接触点保持跟踪。为了促进代理和访问者之间的交互，通常有利的是，利用其他那些与访问者的接触点来帮助识别协同浏览服务中与客户相关联的、该代理所要参与的协同浏览会话。

图 3 示例了当访问者需要帮助（或者代理或自动规则感知到访问者可能需要援助）时如何发起协同浏览会话的示例。在图 3 所示的示例中，代理和访问者通过语音呼叫一起交谈或参与到聊天中，并且网站具有能让访问者发起协同浏览会话的按钮 300。具体地，如果访问者点击按钮 300 以发起会话（图 3A），弹出框 310 将向访问者提供随机的会话 ID 315（号码或字母）。访问者在通话中向代理口头给出号码或者将号码键入到聊天中。然后，代理将该号码提供至协同浏览网络服务器，其中该号码使得协同浏览网络服务器能够将代理连接到正确的协同浏览会话上。

图 4 示例出一种备选的滑块或标签，其中网站上布置有不显眼的按钮 400 并且访问者使用该按钮来发起协同浏览会话。访问者可以点击帮助按钮 400 来寻求帮助，或者代理可以指导访问者找到该按钮并点击它。点击该按钮将开启协同浏览会话，并显示如图 5 所示的弹出框 500。弹出框 500 与图 3 所示的弹出框 310 相同或相似。然后，访问者可向代理文本式输入代码或者告诉代理代码。该代码可通过协同浏览服务来产生或者在访问者浏
览器中产生。除非访问者端和代理侧均得到另一种标识符，否则在任一种情况下，这与会话相关并且需要从访问者传送给代理。

图 6 说明了当代理通过协同服务所提供的界面加入协同会话时所需做的屏幕。如图 6 所示，当代理加入协同会话时，代理会点击他们计算机上将显示图 6 所示的弹出框的图标。然后，代理输入由访问者提供的号码（参见图 3B 和图 5），以加入会话。如图 6 所示，弹出框包括字段 600，其中代理可以将访问者密钥键入该字段中以使该代理加入与访问者的协同浏览会话。

图 7 说明了另一种将号码键入弹出框的备选方案。具体地，在图 7 中协同服务界面旁代理提供了该代理可加入的协同浏览会话（该代理凭借许可、技能设置或一些其他标准所访问的协同浏览会话）的列表。代理从列表中选择协同浏览会话或者在搜索字段 700 中键入会话密钥或其他信息，以便以更快速地从一组可获得的协同浏览会话中定位正确的协同浏览会话。

在本实施例中，代替向访问者提供标识符然后再要求访问者向代理提供标识符的做法，提出让访问者输入可用于让代理找到访问者的协同浏览会话的标识信息。例如，如图 8 所示，当访问者点击标签以发起协同浏览会话时，该标签可给出指示访问者输入标识信息的提示 800 以及供访问者输入标识信息的字段 810。代理访问访问者的名称，并通过将访问者的名称键入类似于框 600（参见图 6）的框中来用该名称选择会话，或者用该名称从当前的会话列表（见图 7）中搜索协同浏览会话。尽管图 8 所示的示例要求用户输入他们的名称，但是也可使用其他的信息来代替，例如客户的位置、账户 ID、电子邮件地址或可访问者网络会话内容中获得的其他信息。

代理可以运行客户关系管理 (CRM) 软件，例如可从 Salesforce™获得的 CRM 软件。代理还可以使用 CRM 软件来定位协同浏览会话。图 9 说明了 CRM 软件界面的一部分的截图。如图 9 所示，代理使用由访问者提供的信息可在 CRM 系统中手动地找到领导者人或联系人。卒选地，当访问者将信息输入字段 810 时，如果 CRM 系统中存在访问者的记录，那么领导者 / 联系人 / 屏幕 / 页面可以提醒代理领导者 / 联系人在网站上和 / 或请求帮助。

图 9 说明了当代理键入访问者名称时的 CRM 软件界面的示例。具体地，在该图中显示代理将请求访问者名称，然后在输入字段 900 中键入该访问者名称。如图 10 所示，CRM 软件将填充与搜索相匹配的联系人和 / 或领导者列表 1000。代理可从该列表（例如，Vera Visitor）中选择条目之一，以提取出 CRM 系统内该访问者的联系记录。

图 11 说明了所选访问者的联系记录的示例。如图 11 所示，联系记录包括供代理点击以加入与访问者的协同浏览会话的按钮 1100。还提供了屏幕共享按钮 1110，以使代理使用屏幕共享会话来与访问者联络，从而使代理能够将信息显示给访问者。通过 CRM 记录来加入协同浏览会话的做法使得 CRM 系统能够在其他联系数据的背景下跟踪协同浏览会话，从而能够根据协同浏览会话信息建立分析。

任选地，CRM 系统可以自动创建新的联系记录或针对新访问者的其他 CRM 记录。CRM 系统与屏幕共享系统间的集成实施方案更详细地记载于 2012 年 9 月 23 日提交的名为 “Integrating Screen Sharing with Customer Relationship Management” 的美国专利 No. 18/624,951 中，其全部内容通过引入并入本文。在引入专利中，使用一种机制来使得 CRM 系统能够检测屏幕共享会话的发起并能够对屏幕共享服务轮询有关屏幕共
享会话的信息，例如屏幕共享会话的时长和关于屏幕共享会参与者的信息。类似机制可用于使得 CRM 系统能够捕捉协同浏览器会话。具体地，CRM 系统可以确定已经选择出代理来参与协同浏览器会话。稍后，CRM 系统可以轮询协同浏览器服务器以获得关于协同浏览器会话的详细内容，例如协同浏览器会话的时长、协同浏览器会话的参与者、在协同浏览器会话期间所访问的页面以及协同浏览器服务可获得的任何其他信息。同样地，出于培训以及审计的目的，可以对协同浏览器会话进行记录。

图 12 展示了另一实例，其中 CRM 软件具有能显示网站的访问者列表的单独标签。在本实施例中，CRM 系统从 CRM 记录提取信息和附加字段，并以列表的形式呈现可获得的协同浏览器会话。例如通过突出显示正请求帮助的访问者和 / 或将这些访问者分选至顶部来对列表进行优化。例如，图 12 所示，提供了网站的访问者列表 1200。其中一个访问者 1202 被列为匿名，这是因为 CRM 系统不具有关于该访问者的任何信息和 / 或该访问者没有向网站提供标识信息。

一些网站提供的聊天式（基于文本的）客户支持。访问者习惯了被询问提供与开启聊天会话有关的识别信息。有关开启聊天所发生的初始信息收集将在本文中称为预聊调查。

能识别协同浏览器会话并将其链接于聊天会话的特定方式取决于协同浏览服务与聊天服务之问的集成水平。如果聊天与协同浏览服务之间进行整合，来自预聊调查的信息可用于执行对匹配协同浏览器会话的查找。图 13A 和 13B 显示了预聊调查窗口的示例，其中当访问者选择经由聊天会话来进行客户服务时该预聊调查窗口可用于从访问者收集信息。

图 14A 和 14B 所示，代理会看到聊天应用程序中的访问者信息，并采用与图 9-12 相关的上述机制，代理可使用该访问者信息来查找协同浏览器会话。如果不使用预聊调查来收集访问者名称或其他识别信息，那么这种信息可以在聊天会话期间由代理收集，并用于定位与访问者相关联的协同浏览器会话。

图 15A 和 15B 显示了当聊天和 CRM 系统与协同浏览服务集成在一起时可使用的示例性 CRM 界面。图 15A 所示，当代理与访问者进行聊天会话时会自动显示针对访问者的联系记录。在本实施例中，CRM 记录包括将代理连接到与访问者相关联的协同浏览器会话的协同浏览按钮 1500。图 15 也示出了有关访问者的代理状态信息，例如访问者已处在网站 1510 的时间量以及访问者在网站 1520 中所位于的页面。这允许代理知道访问者所处于的交易，以便代理能更好地了解访问者可能会遇到问题。

图 15B 显示了当访问者在网站中未被 CRM 系统追踪时的代理屏幕。如图 15B 所示，如果代理想加入与访问者的协同浏览会话，那么该代理被提示输入例如协同浏览密码（参见图 3B）、访问者名称等鉴别信息。

图 16 显示了 CRM 系统与聊天系统之间未进行集成但 CRM 系统的确知道可获得的协同浏览会话的情形。如 16 所示，代理可使用预聊调查所提供的信息或聊天过程中所得到的信息在 CRM 系统中查找针对访问者的联系记录。在图 16 所示的示例中，访问者位置 (Denver) 用于搜索现有的协同浏览会话列表。未选中，代理也可以使用访问者名称或其他标识信息来搜索潜在的匹配协同浏览会话。

图 17 显示了用于聊天服务的示例性代理屏幕。如图 17 所示，代理具有能通过聊天代理屏幕中的按钮 1700 来加入协同浏览会话的能力。聊天服务提供这种代理视图，以使
得代理无需求访问者做任何操作就能在聊天会话中与访问者交互。在图17所示的实施例中，代理可点击按钮1700来自动加入与参与到聊天会话中的访问者所关联的协同浏览会话中。代理不希望访问者操作就能发起协同浏览会话。任选地，代理可以为访问者提供参与协同浏览会话的机会，并向访问者提供接受协同浏览会话的能力。

在一个实施例中，为了使聊天服务能跟踪访问者到访网站，聊天服务将包括该网站上的脚本。本聊天服务的脚本将促使访问者浏览器定期地轮询聊天服务，以确定是否出现了状态改变。例如，聊天服务中的规则会促使聊天服务为网站提供用于询问访问者是否愿意发起聊天会话的弹出框或覆盖框。

聊天服务具有将附加的脚本插入到访问者浏览器中的能力。根据实施例，通过聊天服务将协同浏览脚本下载到访问者端，以便能够发起协同浏览会话。协同浏览脚本可基于聊天服务的规则来进行下载，例如，用户随时可以将选择的条目加入到购物车中，或者聊天代理何时可以选择协同浏览按钮来开启与访问者的协同浏览会话。可以使用多种方法来触发协同浏览脚本的下载，从而向访问者提供协同浏览脚本。任选地，将由协同浏览服务器用来识别所述协同浏览会话的信息（例如，访问者ID或会话ID）可以包括在被下载到访问者端的协同浏览脚本中。因此，聊天服务不需要开启聊天会话就可以将协同浏览脚本下载至访问者。

在另一实施例中，当访问者开启聊天会话时JavaScript（例如，cobrowse.js）被下载并安装到访问者浏览器中。在本实施例中，JavaScript包含指示访问者浏览器轮询具有识别值的会话状态服务器的指令。会话状态服务器可以是聊天服务服务器、协同浏览服务器或网络上的另一服务器。在一个实施例中，识别值唯一与代理和访问者间的聊天会话相关联。

如果代理点击协同浏览按钮1700，则代理协同浏览器将发起协同浏览会话，以创建与识别值相关联的状态。这将导致在轮询服务器端的服务器会话状态发生改变，以便表明协同浏览会话已经开启并已将识别出那个主协同浏览会话的正确的CServer。当访问者浏览器检测到协同浏览会话已开启时，它会向CServer发送将转发给代理的浏览器状态。这能够使代理看到访问者浏览器的视图。等到访问者与会话相连为止，代理再连接到会话中以开始接收数据。

由于JavaScript被下载到与聊天服务所提供的脚本有关的访问者那里，因此不需要访问者采取任何进一步行动（即，访问者不需要点击按钮以表明他们希望代理能够看到其浏览器），协同浏览JavaScript就会促使访问者浏览器对代理可见。为了保护访问者的信息安全性，下载到聊天会话中的JavaScript还包括掩蔽项目列表，以防止与掩蔽字段相关联的敏感信息被转发给协同浏览会话中的代理。

尽管上文描述与基于通信会话的聊天有关，但是可以出现与基于语音的通信会话有关的类似协同浏览会话定位法则加入法。具体地，如果访问者点击按钮（例如，“点击呼叫”按钮）以发起电话呼叫，并输入了电话号码，那么代理可以使用访问者所提供的标识信息来为定位并任选地发起与访问者的协同浏览会话。

轮询最小化

某些情况下，当事人可能希望通过通信会话来交流，例如，协同浏览会话、屏幕共享会话、IP语音会话或者网络会议会话。当一方当事人开启会话另一方当事人却未得知该
会话的情况下，需要向另一方当事人提供信息以便其能够加入该会话。如上所述，多种类型的通信会话可能会遭遇这种问题。尽管将提出与协同浏览相关的示例性解决方案，但是类似的解决方案也可以利用其他形式的通信会话。

【0265】例如，如上面所指出的，代理可以发起与访问者的协同浏览会话，然后让访问者加入协同浏览会话，以便代理能够查看访问者浏览器。在这种情况下，代理可以发起协同浏览会话，以便代理能够查看访问者浏览器上显示的内容。然而，访者不知道代理已经发起了协同浏览会话，因此访者端的JavaScript不知道要开始转发有关访者浏览器内容的信息。

【0266】在这种情况下，访者端的JavaScript可以定期地轮询连接服务器，以确定代理是否已经发起协同浏览会话。对协同浏览网络服务器的连续轮询能够使协同浏览网络服务器周期性地通知访者代理是否已经发起协同浏览会话，以使JavaScript得知主会话服务器的身份，并使访者端的JavaScript开始向协同浏览会话转发更新。

【0267】然而，对于协同浏览网络服务器而言，使用连续轮询（或快速轮询）资源密集性的。例如，如果协同浏览网络服务器隶属于几个网站，每个网站都拥有数以千计的访者，即使每位访者每秒只轮询连接网络服务器一次，那么就算仅处理没有特定协同浏览会话的访者的状态更新，协同浏览网络服务器每秒仍然需要处理成千上万的交易。

【0268】随着网站的可识别访者数量的增长，让访者端JavaScript连续轮询的可用的协同浏览会话是不成比例的。尽管访者端轮询协同浏览网络服务器的快速性可能会下降，但是这在代理选择协同浏览会话的时刻与代理首次开始从访者接收数据的时刻之间引入了潜在的时间滞后。

【0269】有几种方法来解决这个问题。例如，如上面所指出的，访者可以点击与协同浏览服务器相关联的网站中的指定按钮或其他功能（例如，网站上所包括的按钮）。这使得访者能够轮询协同浏览网络服务器，但这不限于网站具有与协同浏览服务相关联的特征的情况。

【0270】根据实施例，协同浏览JavaScript包括询问模块以确定是否已经发起协同浏览会话，其中该询问模块用来捕捉用户的选择并将这些选择解释为轮询协同浏览网络服务器的触发条件。当用户执行特定操作时才让访者端轮询协同浏览的网络服务器的作降低了访者端轮询协同浏览网络服务器的频率，这相应降低了协同浏览网络服务器必须处理的轮询交易。

【0271】例如，假设访者位于某网站中，并且正在得到代理帮助。在这种情况下，该代理点击按钮或其他功能以发起协同浏览会话。为了使访者对该协同浏览会话进行询问并相应地访者能够找到该协同浏览会话，代理可以指示访者在网站上采取不寻常的动作，例如，左击或右击网站的空白区域、摇动其移动设备、让其鼠标转三圈，选择特定字母或其他字符，或采取任何其他不寻常的动作。访者端的JavaScript配置为能查找由代理指定的指定动作。当JavaScript检测到指定动作时，它促使轮询触发模块向协同浏览网络服务器发送询问。协同浏览网络服务器从轮询中接收一些识别信息片段（例如，访者令牌、访者ID或其它值），并搜索与该值相关联的协同浏览会话。如上所指出的，代理在这种方案中得知了访者，并因此当代理人开启协同浏览会话时能够指定访者令牌、访者ID或其它值。
启的协同浏览会话，并向访问者提供识别信息（例如，处理该会话的CServer的身份）。然后，访问者开始将数据发送到协同浏览会话中，以便代理能够查看显示在访问者浏览器上的网页。

[0272] 通过举例的方式，假设访问者登录到网站上，例如登陆到零售商购物车、银行网站，以及服务性、游戏性或使用普遍通用登录的软件（例如，Facebook或Google）。访问者和代理通过聊天或电话来通信，该代理因此得知了访问者的身份。代理可在代理数据库（例如，CRM系统）中找到该访问者，并获得访问者的唯一令牌数据。条选地，令牌可在访问者联系代理以发起会话通信时创建。在任何情况下，代理和访问者均具有相同的令牌。代理发起能向协同浏览服务指定令牌的协同浏览会话。针对与令牌相关联的协同浏览会话的在性，访问者执行“不寻常”动作以促使JavaScript来轮询协同浏览服务。协同浏览服务将轮询与待定的会话相匹配，以将代理和访问者连接到该通信会话（在本实施例中为协同浏览会话）。

[0273] 可检测到的不寻常事件有许多类型。例如，在具有运动传感器的移动设备或其他设备上，浏览器中的JavaScript可使用DeviceMotionEvents来检测摇动事件，这是HTML5工作草案的一部分。不存在具体的“摇动”事件，而是将摇动定义为由DeviceMotionEvents所报告的移动模式。同样地，不所有浏览器都支持该事件，并且不必都以相同方式来处理坐标，而是根据摇动事件能够导致JavaScript来轮询协同浏览网络服务器的方案来决定的。在这种方案中，与访问者交谈或聊天的代理将指示访问者摇动他们的电话或其他设备。当访问者摇动他们的电话时，运行于访问者设备上的JavaScript将轮询协同浏览网络服务器，以使访问者对代理可见。

[0274] 在另一实施例中，访问者和代理甚至不必在摇动事件之前相互通信。在本实施例中，协同浏览JavaScript运行在访问者浏览器中，但协同浏览JavaScript尚未向协同浏览网络服务器发起连接（轮询）。协同浏览会话是针对每个可识别的访问者而发起的，但是代理与协同浏览会话不相关联。然后，已经指示访问者可在网上获取帮助，并且访问者可以摇动他们的电话来征得帮助。当访问者摇动他们的电话/设备时，JavaScript检测到摇动事件并发起与协同浏览网络服务器的连接。这使得访问者能够加入协同浏览会话，并且对网站提供支持的代理能够收到访问者想要帮助的通知。代理被连接到协同浏览会话，并且能看到访问者正在观看的内容。访问者针对网站而被识别，并且这种识别用于还能使访问者和代理使用本文所述的其他方法之一来通过语音或聊天交互。

[0275] 除摇动事件以外的其他突发事件同样可以被捕捉。示例性突发事件（其可由协同浏览JavaScript检测）作为由协同浏览网络服务器发送连接请求的触发条件，包括单击或双击（或者在接触屏上单敲或双敲）网页中不带链接的区域。例如，可以指示访问者点击/双击网站的空白部分。当这种情况发生时，运行于浏览器的JavaScript将捕捉该事件，并解读该事件以引起向协同浏览网络服务器发送的轮询。

[0276] 另一种突发事件可以是在触敏屏上的捏合运动（pinching motion），其随后伴随张开运动。这可指示用户捏合（以使网页缩小），然后张开（以使网页放大）。在例如一秒钟之内的指定时段内的这两个动作可以用作触发条件。

[0277] 另一个突发事件可以是绕圈旋转鼠标。在触敏屏上，当用户触摸屏幕并在屏幕上绘制一系列圆、在屏幕上绘制特定字母、做出对角扫掠动作或者执行那些在正常跳转网站
时很少使用的特定手势时，屏幕上的相似手势都可以被捕捉。

【0278】可以预期，当访问者无意图发起协同浏览会话时也可能偶尔执行异常动作。然而，如果用作触发条件的动作足够不寻常的话，那么相比干让网站的所有访问者定期地轮流协调浏览会话的可用性，转发到协同浏览网络服务器的连接请求的数目将显著减少。

【0279】社交媒体

【0280】例如 Facebook 的社交媒体网站允许用户分享诸如图片、文本和其他感兴趣的内容。每人具有发送器（feed），这是他们添加到其个人页面的项目的运行更新。根据实施例，可以将协同浏览会话的接收链路发布到个人发送器上，这将允许其他访问者链接的人能够将访问者和参与者加入到与访问者的协同浏览会话中。为方便起见，实施方案将以社交媒体网站是 Facebook™ 的方式进行描述，但在这里所描述的概念也可以用于其他社交媒体网站。这实现了许多社交应用，例如社交购物，其中用户从网络获得的网站上进行购物时彼此进行协同浏览。

【0281】在一个实施例中，用户将 Facebook 应用程序安装在他们的 Facebook 配置文件中，这能够将协同浏览会话添加到他们的 Facebook 发送器中。当用户遇到可实现协同浏览的网站（该网站包含协同浏览 JavaScript）时，或者当用户具有包含协同浏览 JavaScript 的插件时，可以向用户提供选项卡以在其 Facebook 发送器上发布协同浏览链接。协同浏览链路包含指向能主管协同浏览会话以及会话标识符（其可用于在协同浏览器中识别协同浏览会话）的协同浏览器的指示。当友好点击协同浏览链接时，启动上述与代理有关的方式来运行的浏览器窗口，以使 Facebook 好友加入协同浏览会话中。其他友好（追随者）也可以加入并关注（该领导者）。如果领导者实现了远程控制，那么追随者可以轮流控制领导者浏览器的去向。

【0282】图 18 示出了社交媒体网站中发布到发送器的示例性发布。如图 18 所示，发布 1800 包括通向协同浏览会话的链接 1810，该链接可被任何访问者发布的人使用以加入与发送器所有者相关联的协同浏览会话中。例如，某人正在购物并想得到有关时尚的建议，那么该人可以向其 Facebook 发送器发布协同浏览邀请，从而在他/她买衣服的同时邀请其友好与其进行协同浏览。尽管图 18 中所示的协同浏览链接 1810 被发布到发送器上，但是可选地可以通过文本消息或电子邮件来将协同浏览器链路 1810 发送给个人或目标团体，以允许那些个人接收所述链路进而加入协同浏览会话中或者在将来的某个时刻加入预定的协同浏览会话中。

【0283】在另一实施例中，在网站上对网站的访问者进行识别，并将其与 Facebook 好友列表、Outlook 联系人列表进行比较。然后，访问者提示他们的 Facebook 好友或 Outlook 联系人之一在同时访问该网站，并且提供与熟人启动协同浏览会话的机会。

【0284】作为另一示例，网站的访问者可被邀请加入到同样在该网站上的其他用户中，从而与这些用户进行协同浏览以实现共同浏览。

【0285】在又一示例中，每当某人把项目放入其购物车中时都可以发起协同浏览会话。在访问者离开网站之后协同浏览会话还可以保留，并且可以向用户发送带有协同浏览邀请的电子邮件或文本消息以供用户返回该网站或返回该用户在该网站内驻留的最后页面。

【0286】在另一实施例中，整合有 Facebook 的网站的访问者可以配备针对同样位于该网站上的好友的侧边栏列表（好友是指其 Facebook 好友）。如果用户点击其好友中的一员
（或者是要加入与其好友的协同浏览会话的选项），则可以向该访问者提供与其好友的协同浏览会话（例如，通过“点击查看我的浏览链接”选项来发起 Facebook 聊天）。

[0287] 协同浏览与屏幕共享集成

[0288] 一些情况下，将从屏幕共享所得到的益处与协同浏览相整合将是有利的。例如，当代理正在查看与访者们的协同浏览会话时，期望向该访者提供该代理实时视频馈频。同样地，代理可能希望将其从与代理的计算机相连的另一设备上获得的视频馈频提供给访问者，该视频馈频例如是来自于相连的平板电脑、智能电话、文档摄像机等的屏幕的实时视频。根据实施例，将为了发起协同浏览会话而在访问者与代理之间完成的连接作为杠杆，以便将两个参与者连接到屏幕共享会话中，其中来自代理的内容经由该屏幕共享会话而提供给访问者。

[0289] 通过提供将屏幕共享会话合并到协同浏览会话的能力，使得当访者者向网站相关的代理发起聊天会话或致电时，产生了不断升级的互动体验。以与客户间的聊天会话 / 电话为开端，支持可升级到协同浏览会话，这使代理能看到网站访问者的网站视图。然后，代理可以使用屏幕共享技术来为访者者提供实况内容，该内容是基于访者者在网站中的位置而为访者者选择的。例如，为了使访者者与代理间具有真人连接，因此可以向访者者提供代理的实时视图，例如网站内的一小插图或者图 3A 和 3B 中的浮动标签 300 的区域 310。代理的实况馈频可以由照相机捕捉并通屏幕共享软件提供给访问者，从而显示给访者者。随着代理和访者者之间的交互的继续，代理可以选择其他内容来提供给访者者。这种附件内容可以是代理运行在其屏幕上。通过屏幕共享会话分享给访问者的演示，或者可以选择的另一实况视频馈频。

[0290] 如上所述，代理借助于协同浏览会话具有看到访者者们的网络浏览器的能力，然而，由于访者者们没有运行屏幕共享软件，代理就不具有看到访者者们屏幕上所出现的其他事物的天生能力。为了使代理能够看到提供给访问者的内容，在一个实施例中，代理会自动作为参与者而加入一个或多个屏幕共享会话中。这允许代理不仅是内容（即，正显示在屏幕共享会话中的内容）的来源，也是该内容的消费者。例如在代理的实况视频正通过屏幕共享会提供给访者者时的情况，这是有用的。通过让代理以访者者身份参与到屏幕共享会话中，可以向代理提供正由访者者提供的视频演示，从而使代理确切知道访者者正在观看的内容。

[0291] 屏幕共享与协同浏览集成在许多场合下都是有用的，这包括销售、客户支持、技术支持等。屏幕共享可以使用 AJAX 查看器。这记载于 2011 年 9 月 22 日提交的名为“Method and Apparatus for Facilitating Participation in a Remote Viewing Session”的美国专利 No. 13/239,757 中，其全部内容通过引用并入本文。

[0292] 使用这种组合的一个优点是，无需访问者下载并安装任何软件就可启动双向网络会议。具体来说就是，协同浏览不需要访问者端的任何下载或 Java / 内存 / Silverlight 或插件。正如有价值性，AJAX 查看器屏幕共享技术无需访问者端的任何下载就能让代理向访者者显示事物。因此，不同于那种两个参与者均需要安装软件的传统双向网络会议，这样的组合无需访问者端的任何下载就让代理进行查看并显示。此外，由于它不依赖于可能不被特定浏览器所支持的 Java 或 Flash，因此它可以用于包括移动设备的任何现代浏览器。

[0293] 尽管提供了将协同浏览会话升级至屏幕共享会话的示例，但是相反的顺序也是可
说明书

以的。 在本实施例中，代理启动与访问者的屏幕共享会话。例如，访问者和代理可以经由聊天会话或语音进行通信，并且该代理可以启动屏幕共享会话以使访问者获得由代理提供的内容。在屏幕共享会话的过程中，为了让代理看到显示在访问者浏览器上的内容，对访问者进行询问。然后，访问者可以利用上述的任何连接方法来发起协同浏览器会话，以向代理显示目前正显示在访问者浏览器上的内容。这使得，完全不需要访问者下载并安装任何软件、插件等，正在帮助访问者的代理既能够通过屏幕共享会话向访问者显示信息，又能够通过协同浏览器会话来查看访问者的屏幕。

[0294] 如上所述，在一个实施例中，代理可以通过打开位于层叠在访问者浏览器页面顶部的 div 中的屏幕共享查看器来向访问者显示信息，并促使该网站的另一部分或代理屏幕上的其他任何事物都在屏幕共享查看器中可见。

[0295] 图 22 显示了根据实施例的协同浏览代理视图 2200 的示例性屏幕截图。如图 22 所示，代理视图包括协同浏览代理视图图标 (header) 2220 中的图标 2210。协同浏览代理视图 2200 还包括显示了访问者浏览器内容的字段 2230。

[0296] 图 23 显示了当代理点击图标 2210 时所产生的弹出框 2310。如图 23 所示，弹出框 2310 包括选择菜单，其中包括允许代理显示代理屏幕的选项 2320 和允许代理查看访问者屏幕的选项 2330。图 24 显示了如果代理选择了显示代理屏幕的选项 2320 时访问者的视图，并且图 25-27 显示了如果代理选择了查看访问者屏幕的选项 2330 时的一系列视图。

[0297] 图 24 显示了如果代理选择了显示代理屏幕的选项 2320 时访问者的视图。如图 24 所示，当代理发起屏幕共享会话并选择要显示代理屏幕的内容时，代理屏幕 2400 的视图将覆盖在访问者的浏览器的顶部 2410 下面。代理浏览器的视图可以包含来自代理所需的任何资源的信息。因此，代理可以选择以包括代理屏幕的视图、视频源（例如所连接设备的 QuickTime 视频源）、来自所连接相机的图像，或其他任何可获得的视频流。尽管未在图 24 中示出，通过选择“显示代理视图”选项，为该代理配备对话框，其中通过所述对话框，允许代理选择将要通过屏幕共享会话提供给访问者的视频源。尽管代理浏览器的视图呈现在位于访问者页面顶部的 div 中，但是屏幕共享视图并未包含在代理的协同浏览查看器中，以避免代理看到自己屏幕上的图像的混淆情况。

[0298] 图 25-27 显示了如果代理选择发起屏幕共享会话（在图 23 中的按钮 2330，在该会话期间访问者屏幕将对代理显示），在访问者端出现的一系列视图。在一个实施例中，开启屏幕共享会话需要利用屏幕共享软件来捕获访问者屏幕。可用于此目的的示例性屏幕共享软件可从 Glance Networks 购得，但本发明并不限于使用 Glance Networks 的屏幕共享软体的实施方案。

[0299] 屏幕共享软件可能需要进行安装。因此，如图 25 所示，如果访问者没有安装屏幕共享软件，那么为访问者配备能提示访问者下载屏幕共享软件客户端的对话框 2500。如果访问者点击链接 2510，则开始下载启动屏幕共享软件客户端。如果访问者已下载过屏幕共享软件客户端，那么图 25 所示的步骤是不必要的，可以跳过。

[0300] 图 26 显示了当屏幕共享软件客户端已下载到访问者端上访问者屏幕的视图，以及何时开启能向代理显示访问者屏幕的屏幕共享会话。如图 26 所示，对访问者提示通知 2600。该通知表明代理已开启屏幕共享会话并使访问者可以允许屏幕共享会话 2610 或者拒绝屏幕共享会话 2620。如果访问者点击“decline（拒绝）” 2620，则屏幕共享会话终止。
如果访者者点击“accept（接受）”2610，列屏幕共享会话发起，并且访者者屏幕将被捕捉并传输给代理以供代理查看。

[0301] 图27示出了在访问者在“accept”2610的情况下屏幕共享会话已开启（其中代理正在查看访问者屏幕）时的访问者自身屏幕的视图。如图27所示，彩色边框2700布置在访问者屏幕周围，以向访问者表明他/她正在与代理分享显示器上所显示的那些内容。还提供了供访者者控制屏幕共享会话的对话框2710。例如，在对话框2710的方框2720中为访者者提供了例如允许代理远程控制访者者计算机的选项、代理停止远程控制的选项、结束屏幕共享会话的选项等。

[0302] 图28显示了当显示访者者屏幕的屏幕共享会话已开启时代理端的访问者屏幕的视图。如图28所示，代理能够查看到与访者者所能看到的内容相同的访问者屏幕内容。不像协同浏览会话过程那样地仅仅简单看到访问者浏览器的内容，使用屏幕共享会话使得访者者看到有关访者者计算机的更多方面，这包括除了访者者浏览器以外的显示方面。此外，由于屏幕共享软件实现了远程控制（其中观察屏幕共享软件的人员能够直接操作主机计算机），使得代理通过使用屏幕共享会话不仅可以看到访者者屏幕而且还可以控制访者者计算机的操作。如上所述，代理进行的远程控制是由访者者授予的特权，因此在屏幕共享会话期间的任何点上访者者都具有恢复其对计算机的控制的选项。

[0303] 秘密协同浏览

[0304] 可以不让访者者知道就让JavaScript自动运行协同浏览会话。这能够使正监控网站的人员可以观察访问者如何使用该网站来确定哪些链接可以找到以及该站点的那些方面可能引起混淆。同样，网站监控者或那些监控网站的代理可以查看访问者的协同浏览会话，以确定特定访问者正在网站上观看什么内容，并在需要时选择与访问者主动进行联系。

[0305] 在一个实施例中，例如聊天服务的服务会随着访者者访问网站而跟踪访者者。在本实施例中，聊天服务（例如Live Person）对具有唯一标识号码（密钥）的访者者进行标示。代理配备有访者者列表以及对这些访问者进行标示的密钥。针对是否有任何开启的会话与该密钥相关联，从网站下载的JavaScript（即，cobrowse.js）促使访者者浏览器来运行服务器。

[0306] 如果代理决定参与到与访者者的协同浏览会话中，那么代理从列表选择访者者或者指示代理系统上的软件发起与该密钥相关的协同浏览会话。然后，该密钥被传递给协同浏览服务器，该服务器会通知服务器已经发起与该密钥相关的协同浏览会话。

[0307] 当访问者输入时，服务器将向访者者返回正值，这将导致cobrowse.js向协同浏览器者提供协同浏览会话的浏览器状态。代理等待直到访者者连接到协同浏览会话，然后代理进行连接。

[0308] 在本实施例中，访者者不采取任何行动来向代理传递信息。相反，跟踪服务器（在本实施例中是LivePerson）向访者者分配跟踪值（密钥）并将该密钥提供给代理。由网站提供的JavaScript用于促使访者者浏览器来运行与该密钥相关的会话，并且类似地协同浏览JavaScript用于促使访者者的网络浏览器参与到协同浏览会话中。访者者不需要做任何操作来参与到协同浏览会话中，访者者也可能不知道他们浏览会话过程中在其浏览器中看到的内容也会被与该网站相关联的代理所看到。这具有使得代理无需用户执行任何操作
就能开启会话的进一步优点，即无需用户点击网站上的特定标签或其他功能。同样，针对营销情报、培训、执法以及其他目的，该服务还可以对会话进行录制。

[0309] 本文所述的方法可以实现为能以控制逻辑来执行的软件，其例如包含在例如为计算机的电子设备的CPU中。本文所述的功能可以实现为存储在非暂时性有形计算机可读介质中的程序指令集合。当以这种方式实现时，该计算机程序包括一组指令，当该组指令由计算机运行时其促使计算机执行能实施上述功能的方法。可编程逻辑可以暂时或永久地安装在非暂时性有形计算机可读介质中，例如只读存储器芯片，计算机存储器，磁盘或其他存储介质。除了以软件来实现之外，本文所述的逻辑可利用分立部件，集成电路，与可编程逻辑设备（例如，现场可编程门阵列或FPGA），结合使用的可编程逻辑，或者包括它们任意组合的任何其他设备来体现。所有此类实施例旨在落入本发明的范围之内。

[0310] 可以在本发明的精神和范围之内进行对本说明书所描述以及附图中所示出的实施例的各种变化和修改。因此，意在指出，包含在上述说明中以及示出在附图中的所有内容应当以说明性而非限制性意义来解释。
图 1
图 14B
图 17
Ed Hardebeck is browsing with Glance

What does anyone think, the red or neon yellow?

Gore Bike Wear Phantom Jacket - Men's
backcountry.com

CoBrowse with me on this website

Like • Comment • Share • Promote • about an minute ago • &

Like • Comment • Share • Promote • about an minute ago • &

图 18
Pilot Program. Give us some info. We'll get you going.

First name: *
Last name: *
Company: *
Email: *

1900

TRY.

Unlike traditional web conferencing, Glance doesn't force nail-biting downloads. Every guest connects instantly, right from their computer, tablet or smartphone. Agents launch sessions "on-the-fly," directly from Salesforce objects or use Salesforce to send invites to future meetings.
图 21
Download and Install Glance

**Glance for Windows**
Version 2.7.3 (920KB)
Requires Microsoft Windows™ 8/7/Vista/XP.
For other PC or Mac versions of Glance, see the Support page.

To install Glance
After clicking the button above:
1. Choose to *Save it to disk*.
2. Choose to *Save In* your Desktop and click *Save*.
3. After the file downloads, double-click the new *GlanceSetup.exe* icon on your desktop.
4. The installation wizard will guide you the rest of the way.
Click here.

Please click the link above.
Debby Mendez would like to view your screen.

Decline

Allow

2600

2620

2610
Glance Cobrowse Integration API