



(12) 发明专利

(10) 授权公告号 CN 102724373 B

(45) 授权公告日 2016. 01. 13

(21) 申请号 201210051779. 0

CN 101557415 A, 2009. 10. 14,

(22) 申请日 2012. 03. 01

审查员 胡莹莹

(30) 优先权数据

2011-047910 2011. 03. 04 JP

(73) 专利权人 佳能株式会社

地址 日本东京都大田区下丸子 3-30-2

(72) 发明人 中岛康辅

(74) 专利代理机构 北京怡丰知识产权代理有限公司 11293

代理人 迟军

(51) Int. Cl.

H04N 1/00(2006. 01)

(56) 对比文件

US 2005108353 A1, 2005. 05. 19,

WO 0158136 A1, 2001. 08. 09,

US 2010106958 A1, 2010. 04. 29,

US 2010110473 A1, 2010. 05. 06,

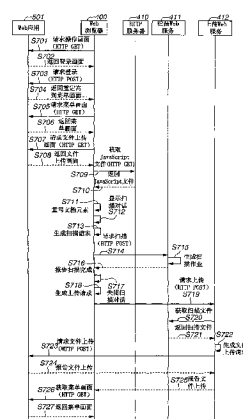
权利要求书2页 说明书11页 附图14页

(54) 发明名称

信息处理装置及信息处理装置的控制方法

(57) 摘要

本发明提供信息处理装置及信息处理装置的控制方法。所述信息处理装置包括被配置为从网络上的 Web 服务器接收画面数据,并基于所述画面数据显示画面的 Web 浏览器。所述信息处理装置存储用来重写由 Web 浏览器接收的画面数据的脚本;并在使用所述脚本重写了所述画面数据之后,根据用户经由基于所重写的画面数据显示的画面发出的指令,执行文件的生成以及所生成的文件的上传。



1. 一种信息处理装置,所述信息处理装置包括:

连接单元,其被配置为与网络连接;

接收单元,其被配置为经由所述网络,从 Web 服务器接收用于在 Web 浏览器上显示第一画面的第一画面数据,所述第一画面是用于上传文件的操作画面;

存储单元,其被配置为存储用于将所述接收单元接收的所述第一画面数据重写为第二画面数据的脚本,所述第二画面数据用于使所述 Web 浏览器能够显示用于接受扫描指令的第二画面;

重写单元,其被配置为使用由所述存储单元存储的所述脚本,将所述第一画面数据重写为所述第二画面数据;以及

控制单元,其被配置为根据依据基于所述第二画面数据显示的所述第二画面接受的用户操作,使扫描单元进行扫描。

2. 根据权利要求 1 所述的信息处理装置,其中,所述扫描单元进行原稿上的图像的扫描,并基于所述图像生成图像文件,并且

其中,所述控制单元根据依据基于所述第二画面数据显示的所述第二画面接受的用户操作,使所述扫描单元进行扫描并生成所述图像文件。

3. 根据权利要求 2 所述的信息处理装置,所述信息处理装置还包括:

上传单元,其被配置为将所生成的图像文件上传到所述 Web 服务器,

其中,所述第二画面还用于接受上传扫描生成的所述图像文件的指令,所述控制单元根据依据基于所述第二画面数据显示的所述第二画面接受的用户操作,使所述上传单元上传所述图像文件。

4. 根据权利要求 1 所述的信息处理装置,其中,所述存储单元存储所述脚本作为所述 Web 浏览器的书签。

5. 根据权利要求 1 所述的信息处理装置,所述信息处理装置还包括:

接受单元,其被配置为从用户接受由所述存储单元存储的所述脚本的执行指令,

其中,响应于所述执行指令的接受而执行由所述存储单元存储的所述脚本的重写。

6. 一种信息处理系统,其包括 Web 服务器和信息处理装置,所述信息处理装置包括:

连接单元,其被配置为与网络连接;

接收单元,其被配置为经由所述网络,从所述 Web 服务器接收用于在 Web 浏览器上显示第一画面的第一画面数据,所述第一画面是用于上传文件的操作画面;

存储单元,其被配置为存储用于将所述接收单元接收的所述第一画面数据重写为第二画面数据的脚本,所述第二画面数据用于使所述 Web 浏览器能够显示用于接受扫描指令的第二画面;

重写单元,其被配置为使用由所述存储单元存储的所述脚本,将所述第一画面数据重写为所述第二画面数据;以及

控制单元,其被配置为根据依据基于所述第二画面数据显示的所述第二画面接受的用户操作,使扫描单元进行扫描。

7. 一种信息处理装置的控制方法,所述信息处理装置包括被配置为与网络连接的连接单元、以及被配置为经由所述网络从 Web 服务器接收用于在 Web 浏览器上显示第一画面的第一画面数据的接收单元,所述第一画面是用于上传文件的操作画面,所述控制方法包括

如下步骤：

存储用于将所述接收单元接收的所述第一画面数据重写为第二画面数据的脚本，所述第二画面数据用于使所述 Web 浏览器能够显示用于接受扫描指令的第二画面；

使用所存储的脚本，将所述第一画面数据重写为所述第二画面数据；以及

根据依据基于所述第二画面数据显示的所述第二画面接受的用户操作，使扫描单元进行扫描。

信息处理装置及信息处理装置的控制方法

技术领域

[0001] 本发明涉及包括从 Web 服务器接收画面数据并基于接收的画面数据显示画面的 Web 浏览器的信息处理装置及信息处理系统。

背景技术

[0002] 已知诸如个人计算机 (PC) 的信息处理装置与网络上的 Web 服务器连接,并在信息处理装置上设置的 Web 浏览器上显示由 Web 服务器提供的操作画面。近年来,包括扫描器和打印机的一些多功能外围设备 (MFP) 设置有上述 Web 浏览器。这种 MFP 在 MFP 的 Web 浏览器上显示由 Web 服务器提供的操作画面,并通过该操作画面从用户接收各种指令。

[0003] 在与这种 MFP 连接时,已知有如下技术:允许用户经由在 MFP 的 Web 浏览器上显示的操作画面发出各种指令,以控制 MFP 的扫描器和打印机。日本特开 2005-149320 号公报讨论了在由 Web 应用提供的操作画面中定义文件上传(遵照请求注释(RFC, Request for Comments)1867)的情况下,MFP 通过响应来自用户的发送指令控制其扫描器,来生成文件,并接着将文件上传到 Web 服务器。

[0004] 根据日本特开 2005-149320 号公报讨论的技术,通用 Web 浏览器需要针对 MFP 来定制,并安装在 MFP 中以使得 MFP 的 Web 浏览器能够控制扫描器。因此,在多个 MFP 针对其中的各个具有不同类型的 Web 浏览器的情况下,需要针对各个 MFP 扩展和安装 Web 浏览器,由此产生开发成本增加的问题。尤其近年来,一些 MFP 已经采用平板 PC 作为其操作单元,因此需要实现该扩展以支持更加多样的 Web 浏览器中的各个。

发明内容

[0005] 本发明旨在提供一种使得能够使用通用 Web 浏览器进行简单的文件上传的构造。

[0006] 根据本发明的一个方面,提供了一种信息处理装置,其包括:连接单元,其被配置为与网络连接;Web 浏览器,其被配置为从所述网络上的 Web 服务器接收画面数据,并基于所述画面数据显示画面;存储单元,其被配置为存储用来重写由所述 Web 浏览器接收的画面数据的脚本;以及控制单元,其被配置为在使用所述脚本重写了所述画面数据之后,根据用户经由基于所重写的画面数据显示的画面发出的指令,执行文件的生成以及所生成的文件的上传。

[0007] 通过以下参照附图对示例性实施例的详细描述,本发明的其他特征和方面将变得清楚。

附图说明

[0008] 包含在说明书中、构成说明书的一部分的附图,示出了本发明的示例性实施例、特征和方面,并且与文字说明一起,用于解释本发明的原理。

[0009] 图 1 示出了根据本发明的示例性实施例的整个信息处理系统。

[0010] 图 2 是示出根据本发明的示例性实施例的 MFP 的配置的框图。

- [0011] 图 3 是示出根据本发明的示例性实施例的服务器的配置的框图。
- [0012] 图 4 示出了根据本发明的示例性实施例的 MFP 的软件配置。
- [0013] 图 5 示出了根据本发明的示例性实施例的服务器的软件配置。
- [0014] 图 6 示出了根据本发明的示例性实施例的 MFP 的画面转变。
- [0015] 图 7 (包括图 7A 和图 7B) 示出了根据本发明的示例性实施例的信息处理系统的处理序列。
- [0016] 图 8 示出了根据本发明的示例性实施例的 HTTP 响应的消息体 (message body) 的示例。
- [0017] 图 9 示出了根据本发明的示例性实施例的小书签 (bookmarklet) 的示例。
- [0018] 图 10 示出了根据本发明的示例性实施例的扫描请求的示例。
- [0019] 图 11 示出了根据本发明的示例性实施例的上传请求的示例。
- [0020] 图 12 示出了根据本发明的示例性实施例的文件上传请求的示例。
- [0021] 图 13 示出了根据本发明的示例性实施例的文件上传完成响应的示例。
- [0022] 图 14 是示出根据本发明的示例性实施例的、当 Web 浏览器通过处理 JavaScript 来重写文档时的处理的流程图。
- [0023] 图 15 是示出根据本发明的示例性实施例的、当 Web 浏览器通过处理 JavaScript 来生成扫描文件时的处理的流程图。
- [0024] 图 16 是示出根据本发明的示例性实施例的、当 Web 浏览器通过处理 JavaScript 来生成上传请求时的处理的流程图。

具体实施方式

- [0025] 以下,将参照附图详细描述本发明的各种示例性实施例、特征及方面。
- [0026] 将描述本发明的第一示例性实施例。在第一示例性实施例中,信息处理系统使用小书签来重写画面数据 (超文本标记语言 (HTML) 数据),以使得 MFP 响应文件浏览按钮的按下,扫描原稿以生成文件,并响应发送按钮的按下,上传生成的文件。
- [0027] 图 1 示出了根据第一示例性实施例的整个信息处理系统。MFP 101 和服务器 102 (Web 服务器) 在局域网 (LAN) 110 上 (在网络中) 以能够通信的方式相互连接。
- [0028] 图 2 是示出 MFP 101 的配置的框图。包括中央处理单元 (CPU) 211 的控制单元 210 控制 MFP 101 的整体操作。CPU 211 读出存储在只读存储器 (ROM) 212 中的控制程序,并进行诸如读取控制和发送控制的各种控制。随机存取存储器 (RAM) 213 用作诸如 CPU 211 的主存储器及工作区的临时存储区域。
- [0029] 硬盘驱动器 (HDD) 214 存储图像数据和各种程序。操作单元接口 (I/F) 215 连接操作单元 219 和控制单元 210。操作单元 219 包括例如具有触摸屏功能的液晶显示单元以及键盘。此外,MFP 101 具有下面描述的 Web 浏览器功能。MFP 101 的 Web 浏览器分析从服务器 102 接收的 HTML 数据,并在操作单元 219 的液晶显示单元上,基于接收的 HTML 数据的描述显示操作画面。
- [0030] 打印机 I/F 216 连接打印机 220 和控制单元 210。从控制单元 210 经由打印机 I/F 216 传送打印机 220 要打印的图像数据,并通过打印机 220 将其打印在记录介质上。扫描器 I/F 217 连接扫描器 221 和控制单元 210。扫描器 221 读取原稿上的图像,生成图像数据

(图像文件),并将生成的数据经由扫描器 I/F 217 输入到控制单元 210。

[0031] 网络 I/F 218 将控制单元 210(MFP 101) 连接到 LAN 110。网络 I/F 218 将图像数据和信息发送到 LAN 110 上的外部装置(例如,服务器 102),并从 LAN 110 上的外部装置接收各种信息。

[0032] 图 3 是示出服务器 102 的配置的框图。包括 CPU 311 的控制单元 310 控制服务器 102 的整体操作。CPU 311 读出存储在 ROM 312 中的控制程序,并进行各种控制处理。RAM 313 用作诸如 CPU 311 的主存储器及工作区的临时存储区域。HDD 314 存储图像数据和各种程序。网络 I/F 315 将控制单元 310(服务器 102) 连接到 LAN 110。网络 I/F 315 从 LAN 110 上的其它装置接收各种信息或者将各种信息发送到 LAN 110 上的其它装置。

[0033] 图 4 示出了 MFP 101 的软件配置。图 4 所示的各个功能单元可以通过包括在 MFP 101 中的 CPU 211 执行控制程序来实现。MFP 101 包括 Web 浏览器 400、超文本传输协议(HTTP) 服务器 410 以及作业控制单元 420。

[0034] Web 浏览器 400 根据 HTTP 协议发送请求并接收响应。接着,Web 浏览器 400 分析接收到的响应,并显示操作画面。Web 浏览器 400 包括 JavaScript 引擎 401 和小书签 402。JavaScript 引擎 401 处理包含在接收的响应中的 JavaScript。Web 浏览器 400 通过处理 JavaScript 能够无需进行 HTTP 通信地重写 HTML 数据,并将 HTTP 请求发送到各种 Web 服务。

[0035] 小书签 402 是登记在 Web 浏览器 400 的书签(bookmark) 中的 JavaScript 应用。当从书签中选择了小书签 402 时,JavaScript 引擎 401 处理小书签 402。

[0036] 当从 Web 浏览器 400 接收到请求时,HTTP 服务器 410 返回例如 HTML 数据或者 JavaScript 文件,或者将来自 Web 浏览器 400 的请求传送到 Web 服务。HTTP 服务器 410 公开的 Web 服务包括扫描 Web 服务 411 以及上传 Web 服务 412。

[0037] 扫描 Web 服务 411 和上传 Web 服务 412 接收来自 Web 应用的 HTTP 请求,并向 Web 应用返回 HTTP 响应。除了 Web 应用以外,有时扫描 Web 服务 411 和上传 Web 服务 412 根据 JavaScript 引擎 401 对小书签的处理,接收从 Web 浏览器 400 发送的请求,并向 Web 浏览器 400 返回响应。

[0038] 扫描 Web 服务 411(第一 Web 服务) 分析接收到的 HTTP 请求,并生成用于控制 MFP 101 的扫描器 211 读取原稿上的图像以生成图像文件作业控制命令。接着,扫描 Web 服务 411 指示作业控制单元 420 控制作业。此外,扫描 Web 服务 411 在接收到作业控制单元 420 的作业完成通知时,返回 HTTP 响应。

[0039] 上传 Web 服务 412(第二 Web 服务) 分析接收到的 HTTP 请求,并生成根据 multipart/form-data 编码所编码的数据。接着,上传 Web 服务 412 将生成的数据作为请求发送到 Web 应用,并从 Web 应用接收发送完成画面。在本示例性实施例中,上传 Web 服务 412 接收从扫描 Web 服务 411 发送的扫描图像,并将图像上传到 Web 应用。上传 Web 服务 412 返回发送完成画面作为 HTTP 响应。

[0040] 作业控制单元 420 通过处理指示的作业控制命令来进行例如打印机 220 的打印处理、扫描器 221 的读取处理或者经由网络 I/F 218 的发送处理。此外,当完成作业处理时,作业控制单元 420 将作业处理完成通知给指令源。

[0041] 图 5 示出了服务器 102 的软件配置。包含在服务器 102 中的 CPU 311 执行控制程

序,从而能够实现图 5 所示的各个功能单元。服务器 102 包括 HTTP 服务器 500 和数据库 510。HTTP 服务器 500 接收来自 Web 浏览器 400 的请求,接着返回例如 HTML 数据或者 JavaScript 文件,或者将来自 Web 浏览器 400 的请求传送到 Web 应用 501。

[0042] Web 应用 501 响应来自 Web 浏览器 400 的请求,生成要在 Web 浏览器 400 上显示的 HTML 数据。此外,Web 应用 501 处理表单 (form) 验证请求,并发出表示会话信息的 cookie。此外,Web 应用 501 处理文件上传请求并将数据登记在数据库 510 中。将生成的 HTML 数据和 cookie 作为 HTTP 响应发送到 Web 浏览器 400。

[0043] 数据库 510 管理 Web 应用 501 的验证信息以及从 Web 浏览器 400 上传到 Web 应用 501 的数据。

[0044] 根据本示例性实施例的信息处理系统通常以上述方式配置。接下来,简述根据本示例性实施例的系统的处理。

[0045] 图 6 示出了根据本示例性实施例的信息处理系统中的 Web 浏览器 400 的画面转变的示例。首先,描述 Web 浏览器 400 的画面配置。如图 6 的操作画面所示,Web 浏览器 400 包括标题栏、统一资源定位符 (URL) 栏、书签按钮以及内容显示区域。标题栏是用于显示 HTML 数据的标题的区域。URL 栏是用于输入或者显示显示的内容的 URL 的区域。书签按钮是用于调用预先登记并存储在 Web 浏览器 400 中的书签的按钮 (用于接收来自用户的执行指令的按钮)。

[0046] 在图 6 所示的示例中,JavaScript 应用以“扫描”的名称登记在书签中。登记在书签中的 JavaScript 应用被称为小书签。下面参照图 9 描述在小书签中描述的 JavaScript 的具体内容。内容显示区域是用于显示指定的 URL 目的地的内容的区域。

[0047] 接下来,描述各个操作画面的内容以及画面转变。在图 6 所示的操作画面中,操作画面 601 至 604 是 Web 浏览器 400 通过处理从 Web 应用 501 返回的内容 (画面数据) 显示的操作画面。另一方面,操作画面 613 至 615 是 Web 浏览器 400 通过处理小书签以重写从 Web 应用 501 返回的内容 (画面数据) 显示的操作画面。

[0048] 登录画面 601 是用于输入用户标识 (ID) 和密码,并请求 Web 应用 501 验证的画面。登录画面 601 包括用户 ID 栏、密码栏以及登录按钮。用户 ID 栏接收 Web 应用 501 验证所需的用户 ID 的输入。密码栏接收 Web 应用 501 验证所需的密码的输入。登录按钮是用于请求 Web 应用 501 基于输入的用户 ID 和密码进行验证的按钮。当按下登录按钮时,只有 Web 应用 501 已经验证成功时,Web 浏览器 400 才将画面切换到菜单画面。

[0049] 菜单画面 602 是用于使用户能够利用由 Web 应用 501 提供的文档管理功能的菜单画面。Web 应用 501 提供的文档管理功能包括文件夹生成功能、文件上传功能以及文件删除功能。菜单画面 602 包括各个功能的按钮以及退出按钮。此外,菜单画面 602 显示由 Web 应用 501 管理的文档文件夹的列表以及文档文件的列表。当按下任意功能按钮时,Web 浏览器 400 将画面切换到与选择的功能相对应的操作画面。另一方面,当按下退出按钮时,Web 浏览器 400 将画面切换到登录画面 601。

[0050] 文件上传画面 603 是用于上传文件的操作画面。操作画面 603 包括文件名输入栏、文件浏览按钮、上传按钮以及返回按钮。当按下文件浏览按钮时,Web 浏览器 400 显示文件选择画面 604。另一方面,当按下上传按钮时,Web 浏览器 400 根据 multipart/form-data 编码对使用浏览按钮选择的图像数据进行编码,并将编码后的数据上传到 Web 应用 501。

[0051] 当上传处理完成时, Web 浏览器 400 将画面切换到菜单画面 602。类似地, 当按下返回按钮时, Web 浏览器 400 将画面切换到菜单画面 602。另一方面, 当按下书签按钮时, Web 浏览器 400 处理小书签并将画面切换到扫描 / 上传按钮 613。

[0052] 文件选择按钮 604 是显示用于使用户能够选择存储在 MFP 101 的 HDD 214 中的图像数据的文件选择对话的操作画面。文件选择对话包括文件列表、确定按钮以及返回按钮。当用户从文件列表中选择文件名并按下确定按钮时, Web 浏览器 400 将画面切换到操作画面 603。接着, Web 浏览器 400 在操作画面 603 的文件名输入栏中显示在文件选择对话上选择的文件名。另一方面, 当按下返回按钮时, Web 浏览器 400 结束文件选择并将画面切换到操作画面 603。

[0053] 扫描 / 上传画面 613 是用于将扫描生成的文件上传的操作画面。如图 6 所示, 消息“扫描就绪”被添加到文件上传画面 603。Web 浏览器 400 处理小书签, 作为其结果, 显示该消息。此外, 作为处理小书签的结果, Web 浏览器 400 还改变当按下文件浏览按钮以及上传按钮时进行的处理的内容。在这种情况下, 当按下浏览按钮时, Web 浏览器 400 显示扫描画面 614, 而不显示文件选择画面 604。

[0054] 此外, 当按下上传按钮时, Web 浏览器 400 向上传 Web 服务 412 发送上传请求, 而不向 Web 应用 501 上传文件。当上传 Web 服务 412 报告上传完成时, 画面被切换到菜单画面 602。类似地, 当按下返回按钮时, Web 浏览器 400 将画面切换到菜单画面 602。

[0055] 扫描画面 614 是显示扫描对话的操作画面, 该扫描对话用于通过控制 MFP 101 的扫描器 211 来读取原稿以生成文件。扫描对话包括文件名输入栏、扫描开始按钮以及返回按钮。文件名输入栏接收通过读取原稿生成的文件名的输入。当用户输入文件名并按下扫描开始按钮时, Web 浏览器 400 向扫描 Web 服务 411 发送扫描请求。

[0056] 当扫描 Web 服务 411 报告文件生成完成时, 画面被切换到扫描完成画面 615。另一方面, 当按下返回按钮时, 画面切换到扫描 / 上传画面 613。扫描完成画面 615 是表示原稿的读取以及文件的生成完成的操作画面。当按下返回按钮时, 画面切换到扫描 / 上传画面 613。

[0057] 下面参照图 7 (包括图 7A 和图 7B) 描述根据本示例性实施例的信息处理系统执行的从扫描原稿到文件上传的处理序列。步骤 S701 至 S708 与从登录 Web 应用 501 到显示文件上传画面 603 的处理相对应。步骤 S709 至 S711 与从执行小书签 402 到重写从 Web 应用 501 发送的内容 (画面数据) 的处理相对应。步骤 S712 至 S716 与生成扫描文件的处理相对应。步骤 S717 至 S726 与将文件上传到 Web 应用 501 的处理相对应。

[0058] 在步骤 S701 中, Web 浏览器 400 向 Web 应用 501 发送用于请求获取操作画面的 HTTP 请求。在步骤 S702 中, Web 应用 501 向 Web 浏览器 400 发送返回登录画面 601 的 HTTP 响应。

[0059] 随后, 当在登录画面 601 上按下登录按钮时, 处理进入步骤 S703。在步骤 S703 中, Web 浏览器 400 向 Web 应用 501 发送请求登录的 HTTP 请求。

[0060] 在步骤 S704 中, Web 应用 501 基于从 Web 浏览器 400 发送的验证信息进行验证处理。如果 Web 应用 501 验证成功, 则 Web 应用 501 生成表示会话信息的 cookie。接着, Web 应用 501 向 Web 浏览器 400 发送指示设置生成的 cookie 以及重定向到菜单画面 602 的 HTTP 响应。

[0061] 在步骤 S705 中, Web 浏览器 400 向 Web 应用 501 发送用于请求菜单画面 602 的 HTTP 请求。此时, Web 浏览器 400 根据在步骤 S704 中返回的 cookie 设置指令, 对 HTTP 请求设置 cookie。

[0062] 在步骤 S706 中, Web 应用 501 向 Web 浏览器 400 发送返回菜单画面 602 的 HTTP 响应。Web 应用 501 在确认对从 Web 浏览器 400 发送的 HTTP 请求设置了有效的 cookie 之后, 返回菜单画面 602。如果未对 HTTP 请求设置有效的 cookie, 则 Web 应用 501 以与步骤 S702 相同的方式返回登录画面 601。类似地, 在之后的步骤中, Web 浏览器 400 将设置有 cookie 的 HTTP 请求发送到 Web 应用 501。

[0063] 接着, 当在菜单画面 602 上按下文件上传功能按钮时, 处理进入步骤 S707。在步骤 S707 中, Web 浏览器 400 向 Web 应用 501 发送请求文件上传画面 603 的 HTTP 请求。

[0064] 在步骤 S708 中, Web 应用 501 发送返回文件上传画面 603 的 HTTP 响应。

[0065] 图 8 示出了在步骤 S708 中发送的 HTTP 响应的消息体的示例。在消息体中以 HTML 格式描述了文件上传画面 603。图 8 所示的 HTML 数据表示当按下发送按钮“上传”时, 根据 multipart/form-data 编码对表单构成元素进行编码, 并将其发送到 Web 应用 501 中由“UploadFile”指定的发送目的地。此外, 图 8 示出了要发送的表单构成元素是文件夹名“RECEIPT OF TRAVEL EXPENSE/FEBRUARY 2011”以及利用文件浏览按钮指定的文件。

[0066] 上述步骤与从登录 Web 应用 501 到显示文件上传画面 603 的处理相对应。接着, 当从书签中选择了小书签 402 时, Web 浏览器 400 通过 JavaScript 引擎 401 处理小书签 402, 并执行与步骤 S709 至 S711 相对应的处理。

[0067] 图 9 示出了小书签的示例。小书签以 JavaScript 格式描述。图 9 所示的小书签表示向文档添加源为由 https://192.168.0.1:8443/js/scan.js 指定的 JavaScript 文件的脚本元素(第二脚本)。在该描述中, “192.168.0.1”表示本地主机。此外, 这里的文档是指 Web 浏览器 400 接收的画面数据。

[0068] 在步骤 S709 中, Web 浏览器 400 向 HTTP 服务器 410 发送请求获取 JavaScript 文件的 HTTP 请求。在步骤 S710 中, HTTP 服务器 410 发送返回 JavaScript 文件的 HTTP 响应。

[0069] 在步骤 S711 中, Web 浏览器 400 向文档添加 JavaScript 文件作为脚本元素。从 HTTP 服务器 410 获取的 JavaScript 文件包含当加载 JavaScript 文件完成时重写构成文档的各种元素的处理的定义。

[0070] 在读取 JavaScript 文件之后, Web 浏览器 400 重写当前显示的文件上传画面 603 的文档元素, 以显示扫描/上传画面 613。下面将参照图 14 描述重写文档元素的处理的细节。

[0071] 作为重写文档元素的结果, Web 浏览器 400 能够实现如下处理, 即 Web 浏览器 400 响应扫描/上传画面 613 上的文件浏览按钮的按下, 扫描原稿并生成文件, 并响应发送按钮的按下上传文件。

[0072] 当在扫描/上传画面 613 上按下文件浏览按钮时, 处理进入步骤 S712。在步骤 S712 中, Web 浏览器 400 显示扫描画面 614。

[0073] 接着, 当在扫描画面 614 上按下扫描开始按钮时, 处理进入步骤 S713。在步骤 S713 中, Web 浏览器 400 生成扫描请求。

[0074] 图 10 示出了扫描请求的示例。扫描请求是要发送到扫描 Web 服务 411 的 HTTP

请求。在图 10 所示的示例中,定义由主机“192.168.0.1:8443”及相对统一资源标识符(URI)“/scan/post”指定的、对扫描 Web 服务 411 的请求。请求内容是根据扫描大小的自动选择、全色以及单面读取的设置来扫描原稿,并生成文件名的名称为“FEBRUARY 6.pdf”的文件。

[0075] 本示例性实施例使用仅指定扫描大小、彩色、选择单面或者双面读取以及文件名的扫描请求,然而,本发明不限于这些内容。本发明可以使用指定其它扫描设置、文件格式设置和 / 或存储目的地的扫描请求。

[0076] 此外,本示例性实施例允许用户仅指定如扫描画面 614 所示的文件名,然而,本发明不限于这些内容。可以允许用户经由扫描画面 614 输入各种扫描设置、文件格式设置以及存储目的地。

[0077] 在步骤 S714 中,Web 浏览器 400 向扫描 Web 服务 411 发送扫描请求。在步骤 S715 中,扫描 Web 服务 411 指示作业控制单元 420 生成作业。

[0078] 接着,当从作业控制单元 420 报告作业完成时,处理进入步骤 S716。在步骤 S716 中,扫描 Web 服务 411 发送用于通知 Web 浏览器 400 扫描完成的 HTTP 请求。

[0079] 接着,当在扫描完成画面 615 上按下返回按钮时,处理进入步骤 S717。在步骤 S717 中,Web 浏览器 400 显示扫描 / 上传画面 613。此时,Web 浏览器 400 将在扫描画面 614 中输入的文件名自动设置到扫描 / 上传画面 613 中的文件名输入栏。

[0080] 接着,当在扫描 / 上传画面 613 中按下上传按钮时,处理进入步骤 S718。在步骤 S718 中,Web 浏览器 400 生成上传请求。

[0081] 图 11 示出了上传请求的示例。上传请求是要发送到上传 Web 服务 412 的 HTTP 请求。在图 11 所示的示例中,定义由主机“192.168.0.1:8443”及相对统一资源标识符(URI)“/upload/multipart”指定的、对上传 Web 服务 412 的请求。请求内容是根据 multipart/form-data 编码对由文件名“FEBRUARY 6.pdf”以及文件夹名“RECEIPT OF TRAVEL EXPENSE/FEBRUARY 2011”指定的文件编码,并将其上传到由 URI“https://docs.xxx.yyy/upload/UploadFile”指定的 Web 应用 501。此外,Web 浏览器 400 发送向 Web 应用 501 上传所需的文件上传画面 603 的 cookie 和字符集。

[0082] 在步骤 S719 中,Web 浏览器 400 向上传 Web 服务 412 发送上传请求。将参照图 16 描述 Web 浏览器 400 从生成上传请求到发送上传请求的处理的细节。

[0083] 在步骤 S720 中,上传 Web 服务 412 在指定文件名的同时,向扫描 Web 服务 411 请求文件获取。

[0084] 在步骤 S721 中,扫描 Web 服务 411 返回与对扫描 Web 服务 411 指定的文件名相对应的文件。此时返回的文件是在步骤 S715 中生成的文件。

[0085] 在本示例性实施例中,上传 Web 服务 412 向扫描 Web 服务 411 请求扫描生成的文件,然而,本发明不限于该配置。扫描 Web 服务 411 可以向 HTTP 服务器 410 发送文件,并且上传 Web 服务 412 可以向 HTTP 服务器 410 请求文件获取。作为选择,扫描 Web 服务 411 可以向上传 Web 服务 412 发送扫描生成的文件。

[0086] 在步骤 S722 中,上传 Web 服务 412 生成文件上传请求。

[0087] 图 12 示出了文件上传请求的示例。文件上传请求是以根据 multipart/form-data 编码的编码状态发送到 Web 应用 501 的 HTTP 请求。如图 12 所示,HTTP 请求包含由 Web 浏览

器 400 请求的文件名和文件夹名 (步骤 S718), 以及从扫描 Web 服务 411 返回的文件 (步骤 S721) 的二值数据构成的数据部分。此外, 通过 Web 浏览器 400 请求的字符集 (步骤 S721) 对消息体进行编码。此外, 在请求头中设置 Web 浏览器 400 请求的 cookie (步骤 S721)。

[0088] 在步骤 S723 中, 上传 Web 服务 412 向 Web 应用 501 发送在步骤 S722 中生成的文件上传请求。在步骤 S724 中, Web 应用 501 向上传 Web 服务 412 发送报告文件上传完成的 HTTP 响应。图 13 示出了 HTTP 响应的示例。如图 13 所示, HTTP 响应指示重定向到由 <https://docs.xxyy.yyy/menu> 指定的 Web 应用 501 的菜单画面 602。

[0089] 在步骤 S725 中, 上传 Web 服务 412 将在步骤 S724 中从 Web 应用 501 返回的 HTTP 响应传送到 Web 浏览器 400。该 HTTP 响应与对在步骤 S719 中发送的 HTTP 请求的响应相对应。

[0090] 在步骤 S726 中, 作为分析在步骤 S725 中返回的 HTTP 响应的内容的结果, Web 浏览器 400 向 Web 应用 501 发送请求获取菜单画面 602 的 HTTP 请求。

[0091] 在步骤 S727 中, Web 应用 501 向 Web 浏览器 400 返回菜单画面 602。作为结果, Web 浏览器 400 能够显示菜单画面 602。

[0092] 接下来, 将参照图 14 至 16 所示的流程图描述 Web 浏览器 400 的 JavaScript 处理的流程。Web 浏览器 400 控制 JavaScript 引擎 401 以处理 JavaScript, 由此能够实现图 14 至图 16 所示的所有处理。

[0093] 图 14 是示出当 Web 浏览器 400 处理 JavaScript 以重写文档 (画面数据) 时的处理流程的流程图。当作为处理小书签 402 的结果, 将 JavaScript 加载到文档元素完成时, Web 浏览器 400 开始处理。

[0094] 在步骤 S1401 中, Web 浏览器 400 获取包括在文档中的所有表单元素。接着, Web 浏览器 400 对在步骤 S1401 中获取的所有表单元素重复从步骤 S1402 至 S1414 的处理。

[0095] 在步骤 S1402 中, Web 浏览器 400 确定表单元素的编码类型是否是 multipart/form-data。如果编码类型是 multipart/form-data (步骤 S1402: 是), 则处理进入步骤 S1403。如果编码类型不是 multipart/form-data (步骤 S1402: 否), 则 Web 浏览器 400 针对下一个表单元素重复从步骤 S1402 至 S1414 的处理。

[0096] 在步骤 S1403 中, Web 浏览器 400 获取构成表单元素的元素。那些元素的具体示例可以包括 input 元素、select 元素以及 text-area 元素。接着, Web 浏览器 400 针对在步骤 S 1403 中获取的所有表单构成元素重复从步骤 S1404 至 S1411 的处理。

[0097] 在步骤 S1404 中, Web 浏览器 400 确定表单构成元素是否为具有类型属性 “file (文件)” 的 input 元素。如果表单构成元素是该种 input 元素 (步骤 S1404: 是), 则处理进入步骤 S1405。如果表单构成元素不是该种 input 元素 (步骤 S1404: 否), 则 Web 浏览器 400 针对下一个 input 元素重复从步骤 S1404 至 S1411 的处理。

[0098] 在步骤 S1405 中, Web 浏览器 400 生成表单构成元素 (类型 “file”) 的克隆节点。在步骤 S1406 中, Web 浏览器 400 将克隆节点的类型设为 “hidden”。在步骤 S1407 中, Web 浏览器 400 生成 span 元素。

[0099] 在步骤 S1409 中, Web 浏览器 400 添加类型 “text” 的 input 元素以及类型 “button” 的 input 元素作为在步骤 S1407 中生成的 span 元素的子元素。该按钮元素具有响应按钮的按下显示上述扫描对话的事件处理。

[0100] 接着,在步骤 S1410 中,Web 浏览器 400 用 span 元素替换表单构成元素(具有类型属性“file”的 input 元素)。

[0101] 作为从步骤 S1407 至 S1410 中的处理的执行结果,文件上传画面 603 上的文件浏览按钮能够用显示扫描对话的扫描按钮替换。用 JavaScript 能够输入文件名的文本栏替换文件上传画面 603 中的文件名输入栏。以下,将该 span 元素称为“哑(dummy)节点”。

[0102] 在步骤 S1411 中,将上述克隆节点添加为表单元素的子元素。在对 JavaScript 设置以提供例如表单构成元素的输入检查(如图 8 所示的 check 函数)的情况下,需要添加原始表单构成元素,而不从文件上传画面 603 中删除。

[0103] 以这种方式,重复从步骤 S1404 至 S1411 的处理,使得能够实现重写构成文件上传画面 603 的文件名输入栏和文件浏览按钮。

[0104] 在步骤 S1412 中,Web 浏览器 400 确定是否进行了表单构成元素重写处理。更具体地说,表单构成元素重写处理与重写文件名输入栏和文件浏览按钮相对应。如果文件名输入栏和文件浏览按钮被重写(步骤 S1412:是),则处理进入步骤 S1413。如果文件名输入栏和文件浏览按钮未被重写(步骤 S1412:否)(更具体地说,如果表单构成元素不是具有类型属性“file”的 input 元素),则 Web 浏览器 400 针对下一个表单元素重复从步骤 S1402 至 S1414 的处理。

[0105] 在步骤 S1413 中,Web 浏览器 400 复制表单元素的 action(动作)属性的值,并将其设置为其它属性。在步骤 S1414 中,Web 浏览器 400 将 action 属性的值改变为生成文件上传请求的函数。作为该处理的结果,Web 浏览器 400 被配置为以如下方式操作:当在文件上传画面 603 上按下上传按钮时,Web 浏览器 400 向上传 Web 服务 412 发送通过处理 JavaScript 生成的上传请求。

[0106] 在该处理的执行之前,Web 浏览器 400 被配置为以如下方式操作:当按下上传按钮时,Web 浏览器 400 在根据 multipart/form-data 编码对文件进行编码之后,将其上传到 Web 应用 501。在步骤 S1413 中,将原始 action 属性的值保存为其它属性,因为需要属性以向文件上传请求的请求头“X-UPLOAD-ACTION”设置 Web 应用 501 的发送目的地。

[0107] 以这种方式,重复从步骤 S1402 至 S1414 的处理,能够实现重写文件上传画面 603 中的上传操作。在步骤 S1415 中,Web 浏览器 400 确定是否改变了表单。当 Web 浏览器 400 确定改变了表单时,处理进入步骤 S1416。在步骤 S1416 中,Web 浏览器 400 生成在扫描/上传画面 613 上显示消息的 span 元素。接着,在步骤 S1417 中,Web 浏览器 400 在指定预定位置的同时,向文档添加 span 元素。结果,在扫描/上传画面 613 上出现消息“扫描就绪”。

[0108] 图 15 是示出当 Web 浏览器 400 处理 JavaScript 以生成扫描文件时的处理流程的流程图。当在扫描画面 614 上按下扫描开始按钮时,Web 浏览器 400 开始该流程图中的处理。

[0109] 在步骤 S1501 中,Web 浏览器 400 确定是否在扫描画面 614 的文件名输入栏中输入任意文件名。如果在扫描画面 614 中输入了文件名(步骤 S1501:是),则处理进入步骤 S1502。如果在扫描画面 614 中未输入文件名(步骤 S1501:否),则处理进入步骤 S1512。在步骤 S1512 中,Web 浏览器 400 显示通知用户未输入文件名的警告消息,接着,结束处理。

[0110] 在步骤 S1502 中,Web 浏览器 400 生成图 10 所示的扫描请求。接着,在步骤 S1503 中,Web 浏览器 400 向扫描 Web 服务 411 发送扫描请求。

[0111] 在步骤 S1504 中,Web 浏览器 400 确定是否从扫描 Web 服务 411 接收到扫描完成响应。如果 Web 浏览器 400 接收到扫描完成响应(步骤 S1504:是),则处理进入步骤 S1505。如果 Web 浏览器 400 未接收到扫描完成响应(步骤 S1504:否),则 Web 浏览器 400 重复步骤 S1504 中的处理。

[0112] 在步骤 S1505 中,Web 浏览器 400 对哑节点的文件名设置在扫描画面 614 中输入的文件名。在步骤 S1506 中,Web 浏览器 400 对克隆节点设置在扫描画面 614 中输入的文件名,接着结束处理。结果,在扫描/上传画面 613 上出现文件名。此外,当按下上传按钮时,通过图 8 所示的 check 函数正常完成文件名输入检查,并调用设置为 action 属性的上传请求生成函数。

[0113] 图 16 是示出当 Web 浏览器 400 处理 JavaScript 以生成上传请求时的处理流程的流程图。当作为按下扫描/上传画面 613 上的上传按钮的结果,调用相应表单元素的上传请求生成函数时,Web 浏览器 400 开始该流程图中的处理。

[0114] 在步骤 S1601 中,Web 浏览器 400 将 Web 应用 501 的发送目的地、字符集以及 cookie 设置到上传请求的 HTTP 头(X-UPLOAD-ACTION, X-UPLOAD-CHARSET, 以及 X-UPLOAD-COOKIE)中。在步骤 S1602 中,Web 浏览器 400 获取构成当前表单元素的元素。那些元素的具体示例可以包括:input 元素、select 元素以及 text-area 元素。

[0115] 接着,Web 浏览器 400 针对在步骤 S1602 中获取的所有表单构成元素重复从步骤 S1603 至 S1606 的处理。在步骤 S1603 中,Web 浏览器 400 确定表单构成元素是否是哑节点的子元素。如果 Web 浏览器 400 确定元素是哑节点的子元素以外的元素(步骤 S1603:是),则处理进入步骤 S1604。如果 Web 浏览器 400 确定元素是哑节点的子元素(步骤 S1603:否),则 Web 浏览器 400 针对下一个表单构成元素重复从步骤 S1603 至 S1606 的处理。

[0116] 在步骤 S1604 中,Web 浏览器 400 确定是否对表单构成元素设置了 name 属性。如果对表单构成元素设置了 name 属性(步骤 S1604:是),则处理进入步骤 S1605。如果未对表单构成元素设置 name 属性(步骤 S1604:否),则 Web 浏览器 400 针对下一个表单构成元素重复从步骤 S1603 至 S1606 的处理。

[0117] 在步骤 S1605 中,Web 浏览器 400 确定表单构成元素是否是克隆节点。如果表单构成元素是克隆节点(步骤 S1605:是),则处理进入步骤 S1606。如果表单构成元素不是克隆节点(步骤 S1605:否),则处理进入步骤 S1616。

[0118] 在步骤 S1606 中,Web 浏览器 400 使表单构成元素的 name 属性的值、表单构成元素的 value 属性的值与根据 value 属性的值确定的 content-type(内容类型)配对,将该对添加到消息体。

[0119] 另一方面,在步骤 S1616 中,Web 浏览器 400 确定表单构成元素是否是具有类型属性“submit”或者“button”的 input 元素。如果表单构成元素是具有类型属性“submit”或者“button”的 input 元素(步骤 S1616:是),则 Web 浏览器 400 针对下一个表单构成元素重复从步骤 S1603 至 S1606 的处理。

[0120] 如果表单元素是其它类型的元素(步骤 S1616:否),则处理进入步骤 S1617。在步骤 S1617 中,Web 浏览器 400 将表单构成要素的 name 属性的值与 value 属性的值配对,并将该对添加到消息体。

[0121] 在步骤 S1607 中,Web 浏览器 400 确定是否对当前按下的 submit 按钮设置了 name

属性。如果未对 submit 按钮设置 name 属性（步骤 S1607：否），则处理进入步骤 S1609。如果对 submit 按钮设置了 name 属性（步骤 S1607：是），则处理进入步骤 S1608。在步骤 S1608 中，Web 浏览器 400 将 submit 按钮的 name 属性的值与 value 属性的值配对，并将该对添加到消息体。

[0122] 根据上述处理，对上传请求设置上传 Web 服务 412 向 Web 应用 501 上传文件时所需的信息。最后，在步骤 S1609 中，Web 浏览器 400 将生成的上传请求发送到上传 Web 服务 412，接着结束处理。

[0123] 以这种方式，在本示例性实施例中，Web 浏览器 400 使用小书签扩展 Web 应用 501 的功能性（重写从 Web 应用 501 接收的画面数据）。结果，变得能够响应文件浏览按钮的按下，扫描原稿以生成文件，并能够响应发送按钮的按下上传文件。

[0124] 在第一示例性实施例中，以 MFP 提供上传 Web 服务 412 的方式配置信息处理系统，然而，本发明不限于该配置。可以以包括 HTTP 服务器的其它服务器提供上传 Web 服务 412 的方式配置信息处理系统。在这种情况下，可以以扫描 Web 服务 411 将扫描生成的文件发送到上传 Web 服务 412 或者包括 HTTP 服务器的其它服务器的方式配置信息处理系统。

[0125] 此外，在第一示例性实施例中，以 MFP 101 提供由小书签获取的 JavaScript 文件的方式配置信息处理系统，然而，本发明不限于该配置。可以以包括 HTTP 服务器的其它服务器可以提供 JavaScript 文件的方式配置信息处理系统。

[0126] 此外，在第一示例性实施例中，以在 MFP 101 中提供 Web 浏览器 400 的方式配置信息处理系统，然而，本发明不限于该配置。信息处理系统可以被配置为利用由信息装置（例如平板 PC 或者智能手机）提供的 Web 浏览器。

[0127] 此外，在第一示例性实施例中，以 MFP 101 提供上传 Web 服务 412 的方式配置信息处理系统。然而，信息处理系统可以被配置为通过使用 JavaScript 控制 Web 浏览器 400 的发送功能来上传文件。在这种情况下，可以从系统配置中省略上传 Web 服务 412。在这种情况下，可以以扫描 Web 服务 411 将扫描生成的文件发送到包括 HTTP 服务器的其它服务器的方式配置信息处理系统。

[0128] 本发明的各方面还可以通过读出并执行记录在存储设备上的用于执行上述实施例的功能的程序的系统或装置的计算机（或诸如 CPU 或 MPU 的设备）来实现，以及通过由系统或装置的计算机通过例如读出并执行记录在存储设备上的用于执行上述实施例的功能的程序来执行各步骤的方法来实现。鉴于此，例如经由网络或者从用作存储设备的各种类型的记录介质（例如计算机可读介质）向计算机提供程序。

[0129] 虽然参照示例性实施例对本发明进行了描述，但是应当理解，本发明并不限于所公开的示例性实施例。应当对所附权利要求的范围给予最宽的解释，以使其涵盖所有这些变型例以及等同的结构和功能。

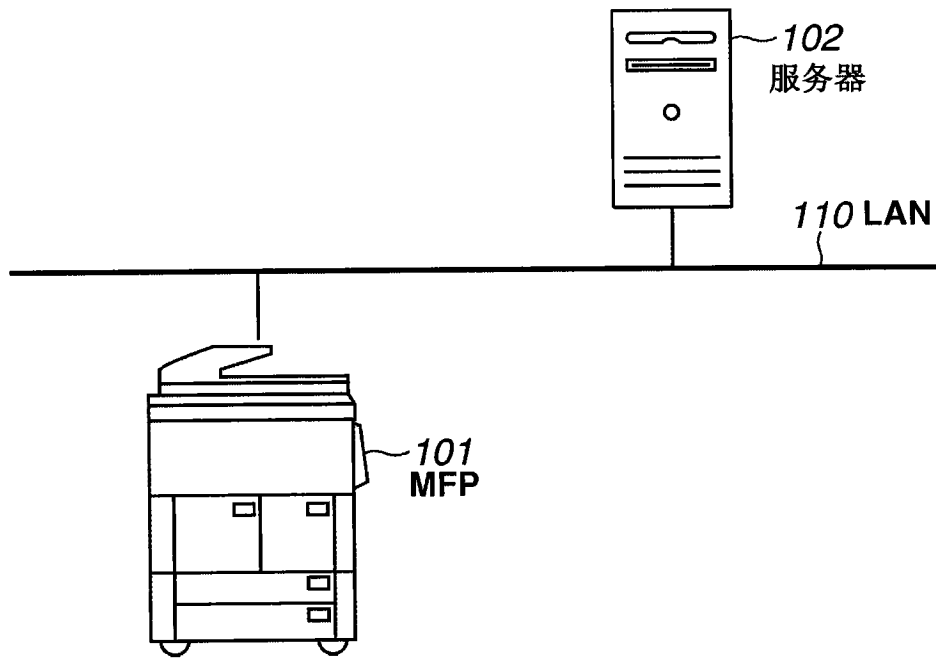


图 1

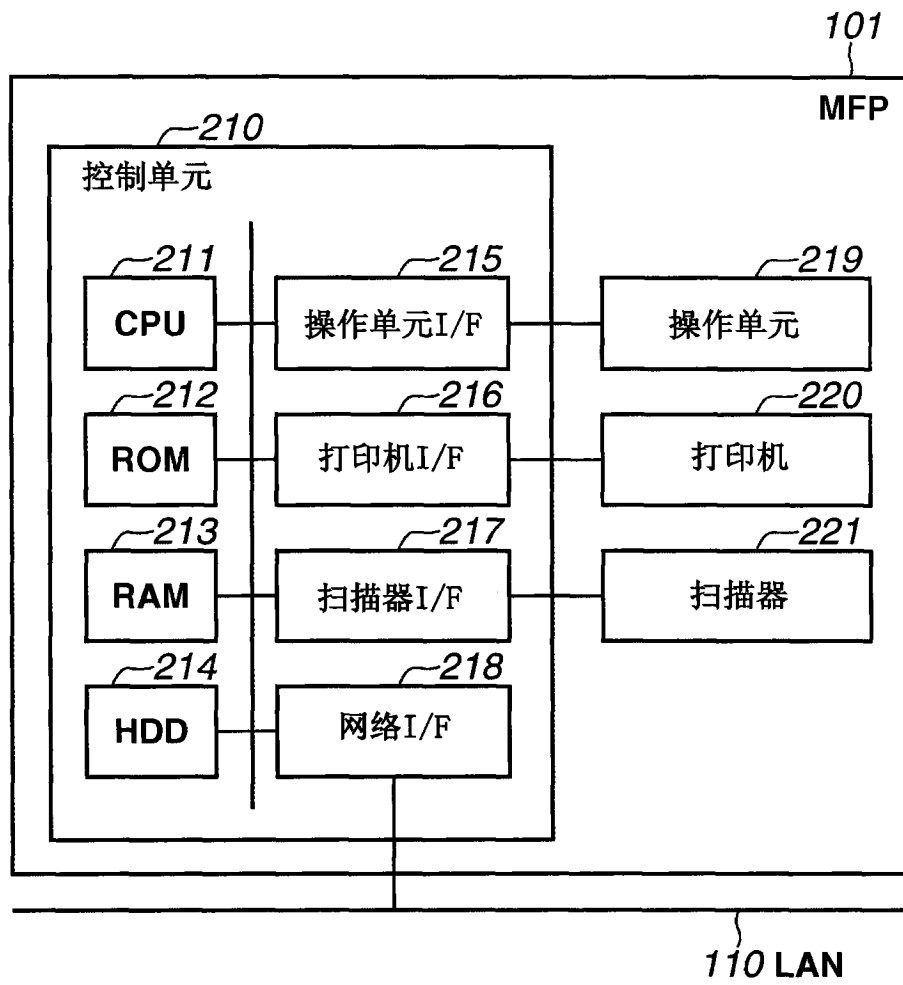


图 2

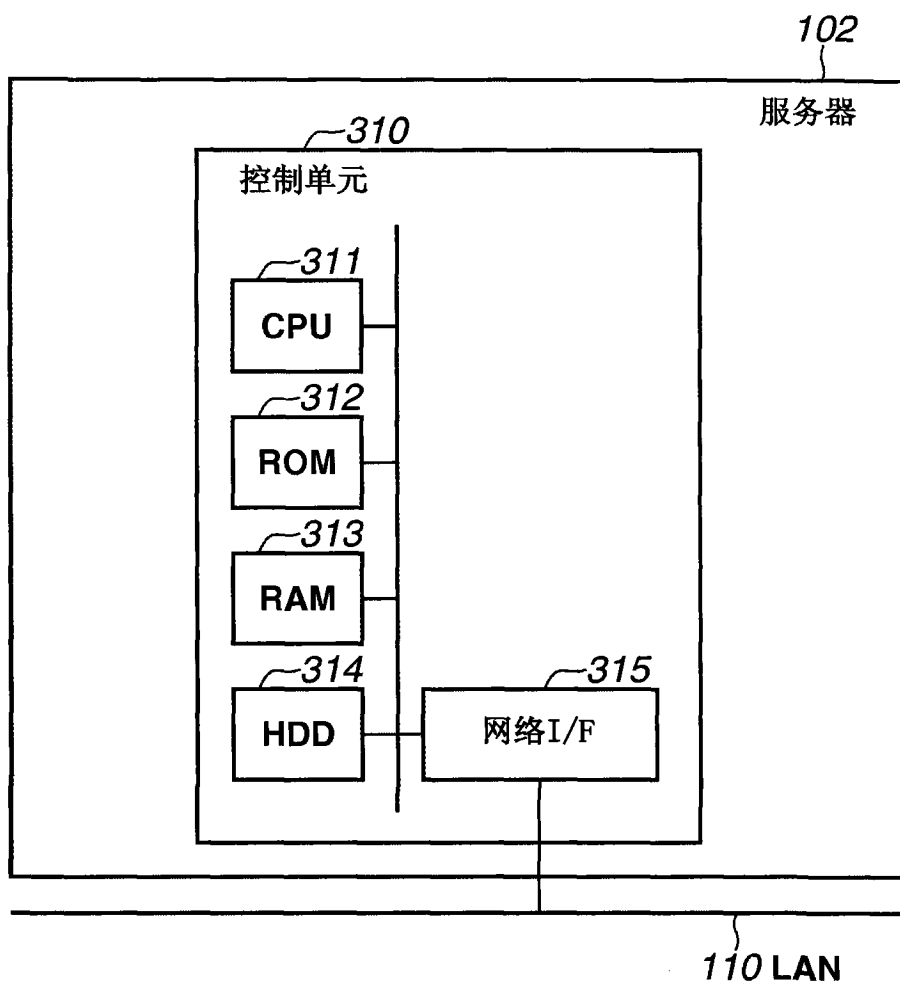


图 3

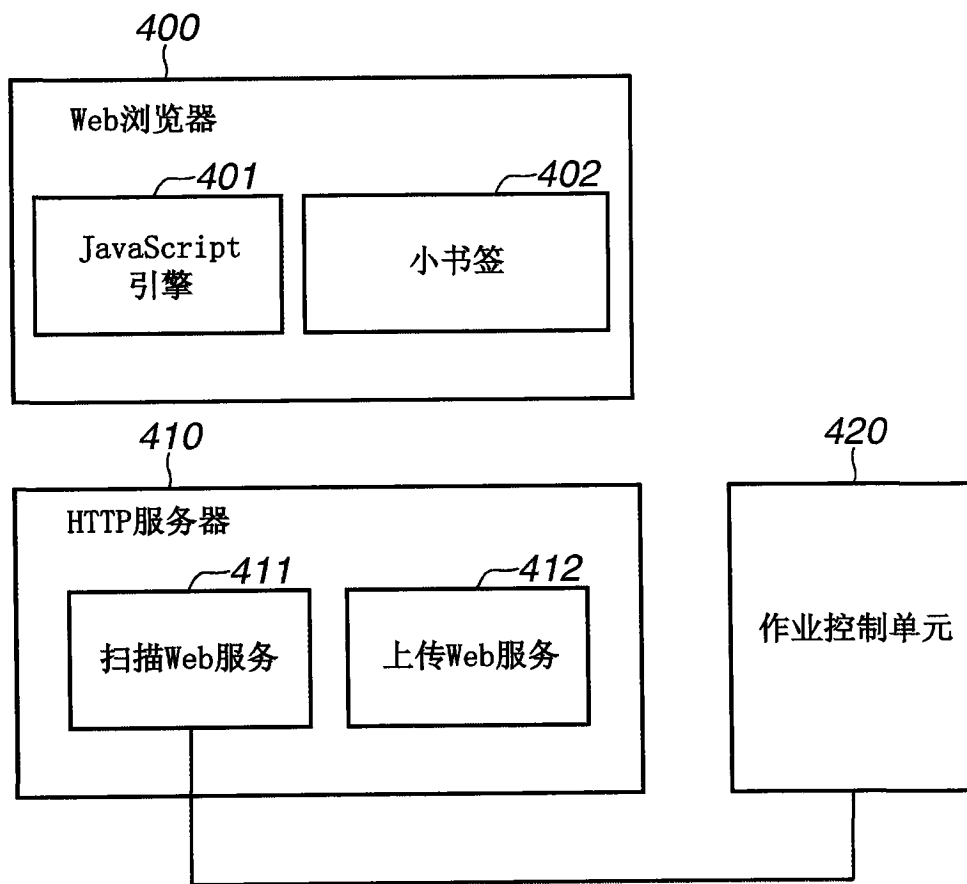


图 4

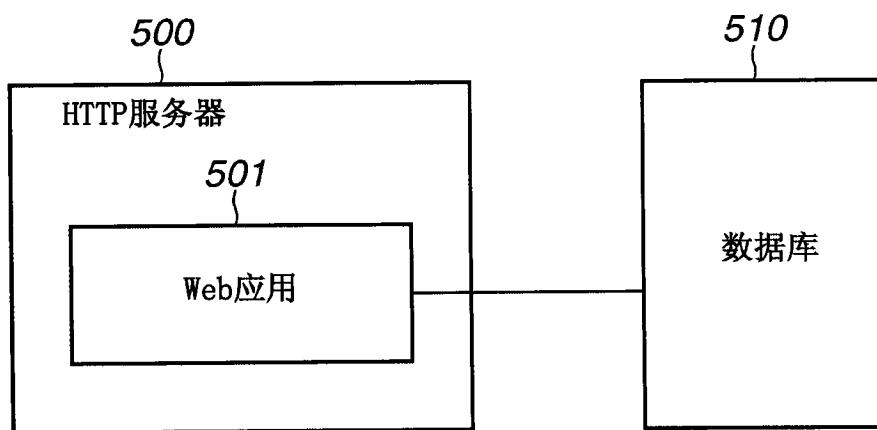


图 5

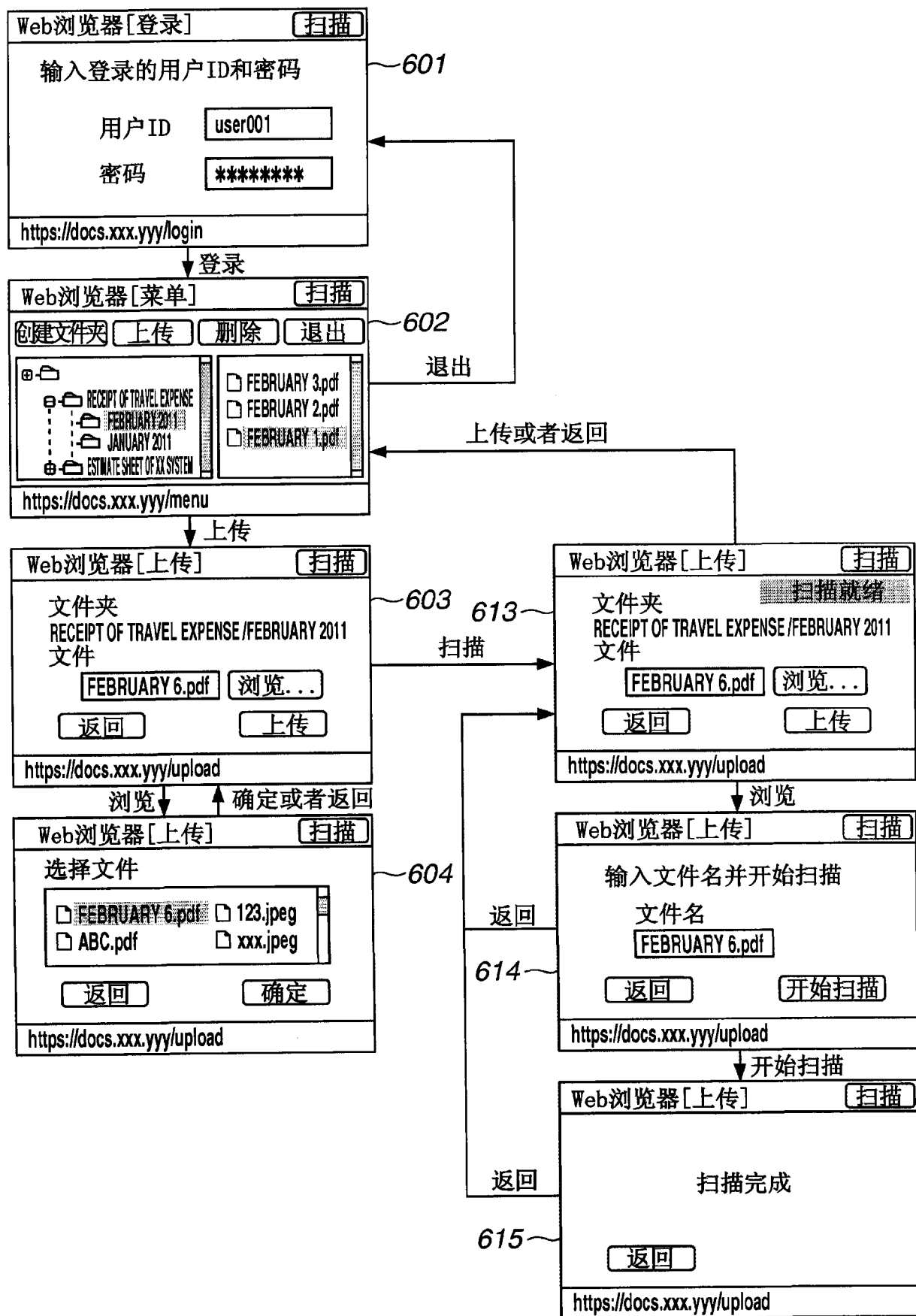


图 6

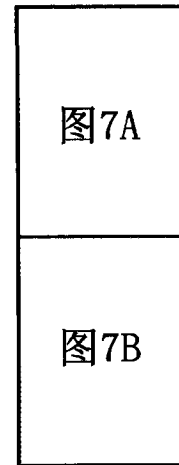


图7

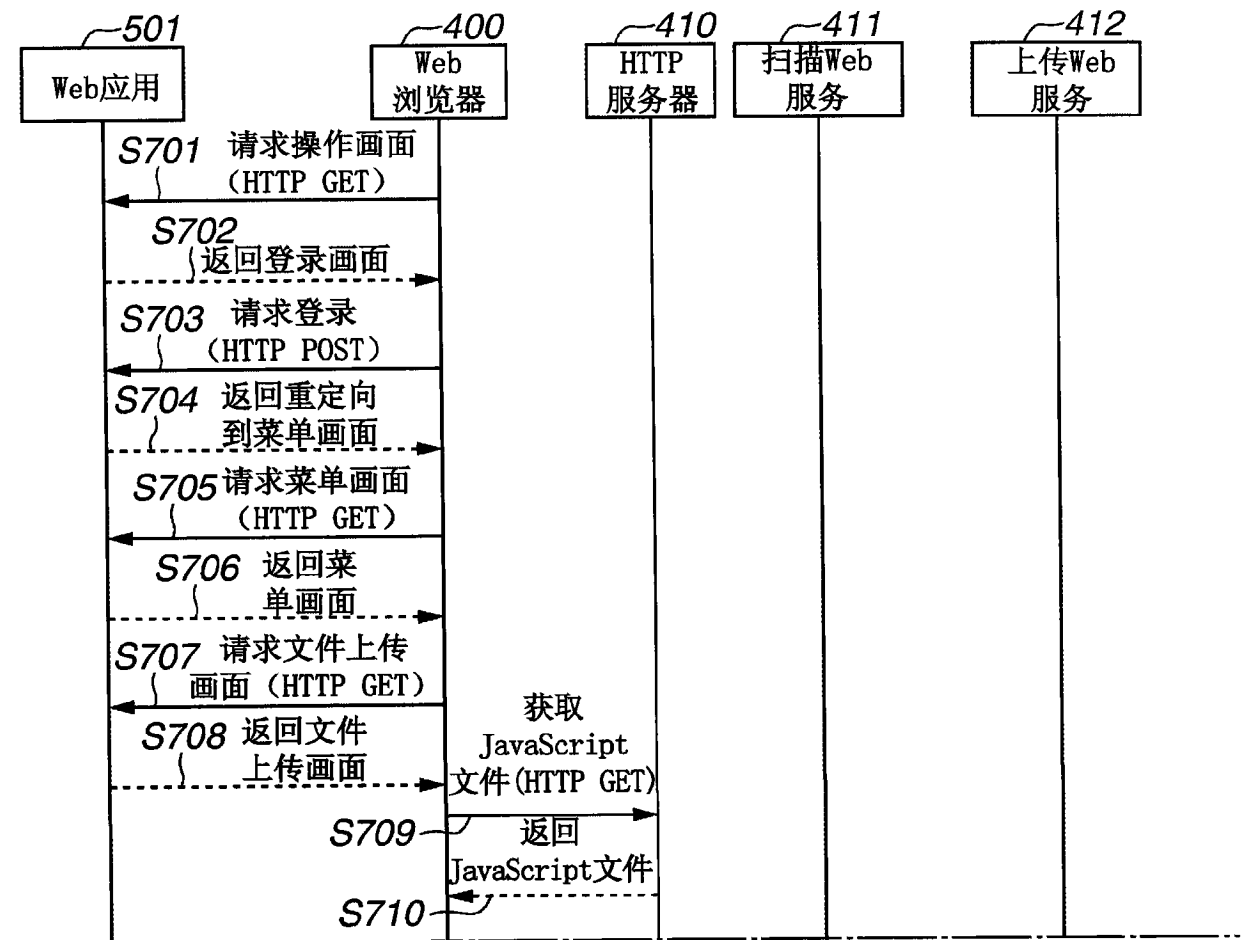


图 7A

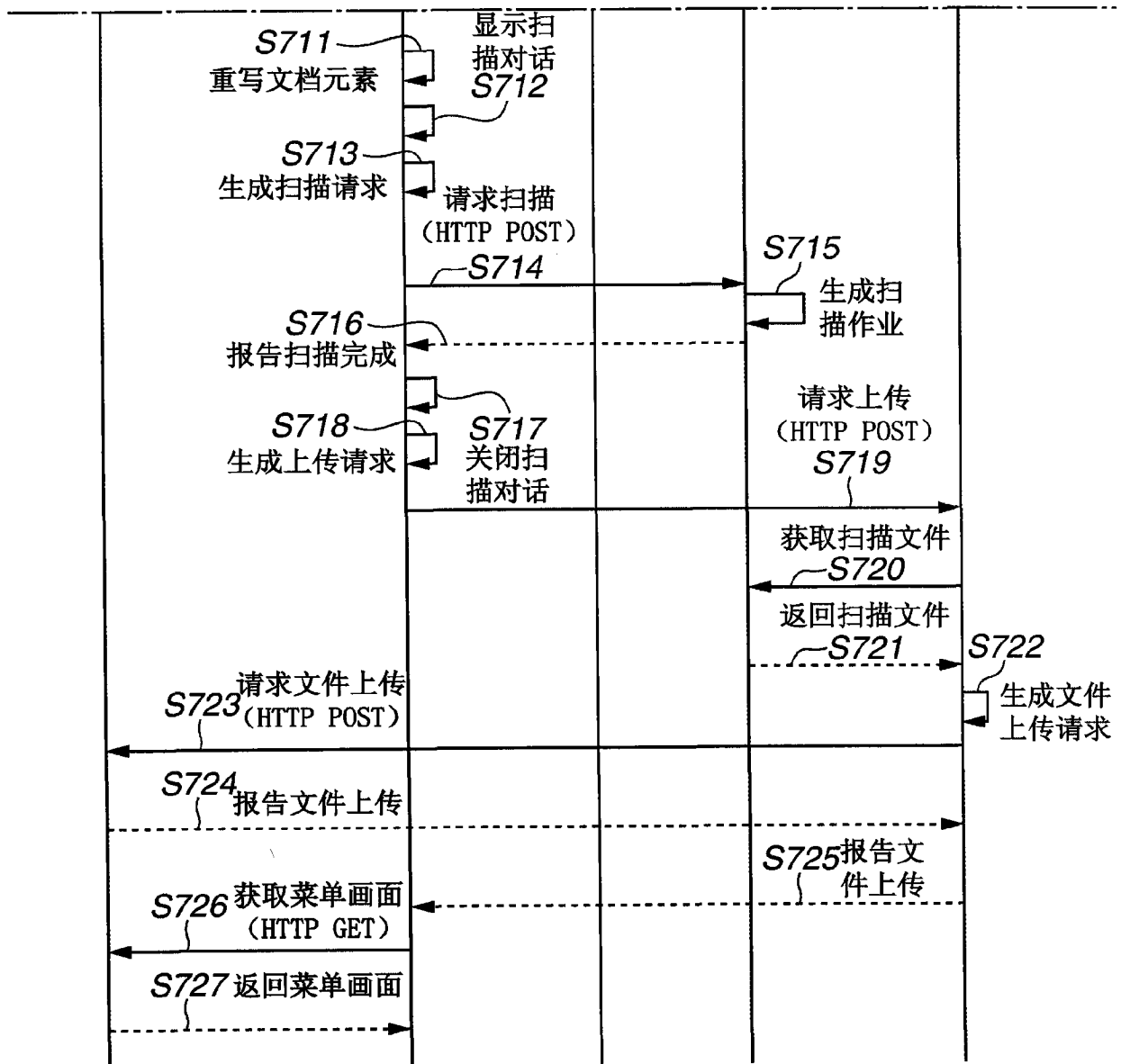


图 7B

```

<html>
<head>
<title>FILE UPLOAD</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<script type="text/javascript">
function check(){
    if(document.form.file.value==""){
        alert(" 未输入文件名 ");
        return false;
    }else{
        return true;
    }
}
</script>
</head>
<body>
<form method="POST" name="form" enctype="multipart/form-data" onSubmit="return check()" action="UploadFile">
<p>FOLDER</p>
<div>RECEIPT OF TRAVEL EXPENSE /FEBRUARY 2011</div>
<p>FILE</p>
<div>
    <input type="file" name="file" size="32">
</div>
<input type="hidden" name="folder_name" value="RECEIPT OF TRAVEL EXPENSE /FEBRUARY 2011">
<p>
    <input type="button" onclick="history.back()" value=" 返回 ">
    <input type="submit" value=" 上传 ">
</p>
</form>
</body>
</html>

```

图 8

```
javascript:if(document.getElementById("scan_js")==undefined)
(function(){
var s=document.createElement("script");
s.src="https://192.168.0.1:8443/js/scan.js";
s.id="scan_js";
document.body.appendChild(s)
}
)();
```

图 9

```
POST /scan/post HTTP/1.1
Host: 192.168.0.1:8443
Content-Length: 97
Content-Type: text/plain; charset=UTF-8

{"size":"auto", "color":"auto", "plex":"simplex", "file":"FEBRUARY 6.pdf"}
```

图 10

```
POST /upload/multipart HTTP/1.1
Host: 192.168.0.1:8443
X-UPLOAD-ACTION: https://docs.xxx.yyy/upload/UploadFile
X-UPLOAD-CHARSET: UTF-8
X-UPLOAD-COOKIE: sessionid=75dbfc6bd0b4e6e35278d901122693c5
{ "file": { "filename": "FEBRUARY 6.pdf", "contentType": "application/pdf", "folder_name": "RECEIPT OF TRAVEL EXPENSE /FEBRUARY 2011" }
```

图 11

```
POST /UploadFile HTTP/1.1
Host: docs.xxx.yyy
User-Agent: Mozilla/5.0 (MFP; IR-S/1.3; like Gecko)
Cookie: sessionid=75dbfc6bd0b4e6e35278d901122693c5

Content-Type: multipart/form-data; boundary=-----309001771318958
Content-Length: 10490

-----309001771318958
Content-Disposition: form-data; name="file"; filename="FEBRUARY 6.pdf"
Content-Type: application/pdf

%PDF-1.4
...
%%EOF
-----309001771318958
Content-Disposition: form-data; name="folder"

RECEIPT OF TRAVEL EXPENSE /FEBRUARY 2011
-----309001771318958--
```

图 12

```
HTTP/1.1 303 See Other
Server: Apache-Coyote/1.1
Date: Sun, 6 Feb 2011 08:27:28 GMT
Location: https://docs.xxx.yyy/menu
Transfer-Encoding: chunked
```

图 13

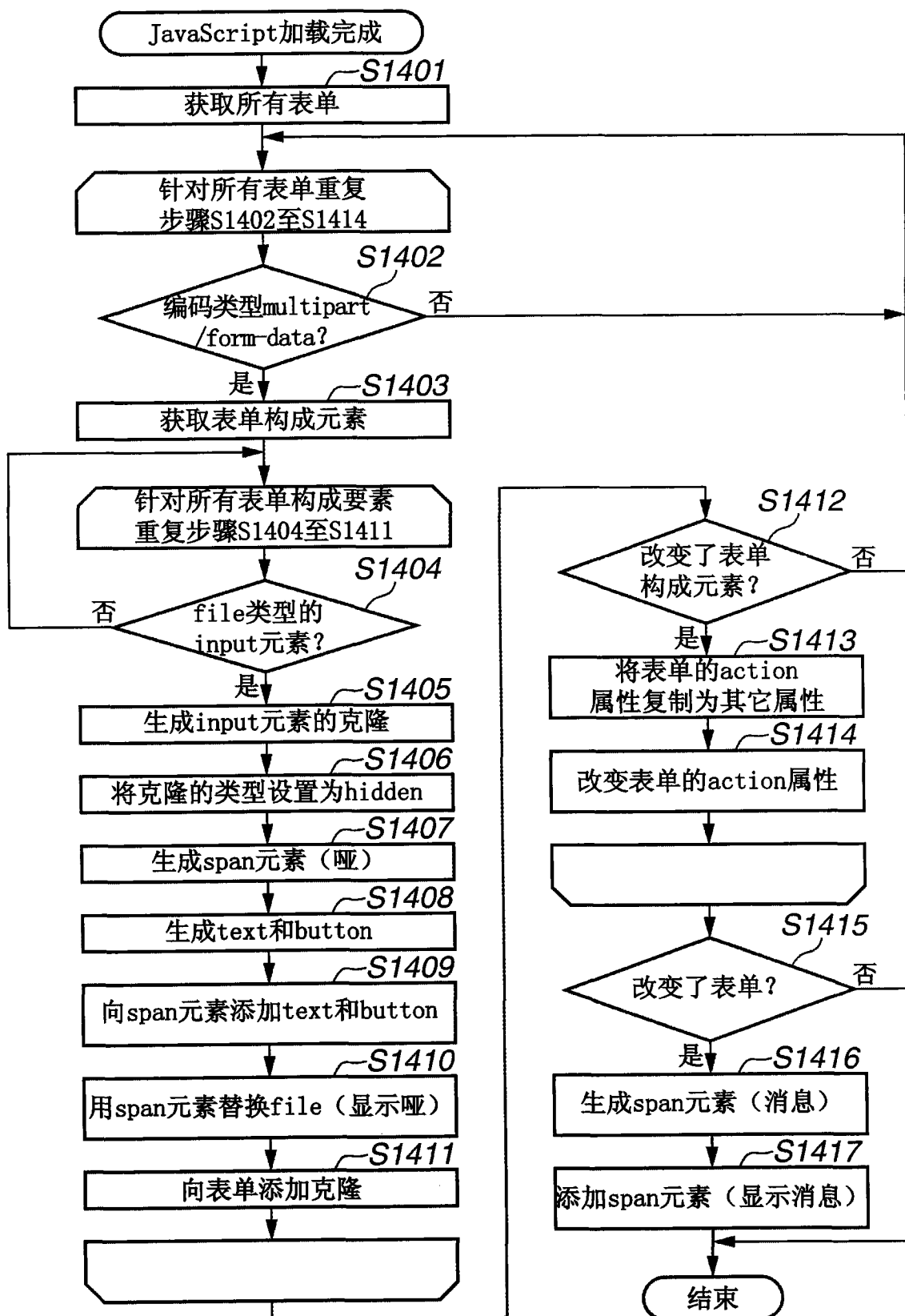


图 14

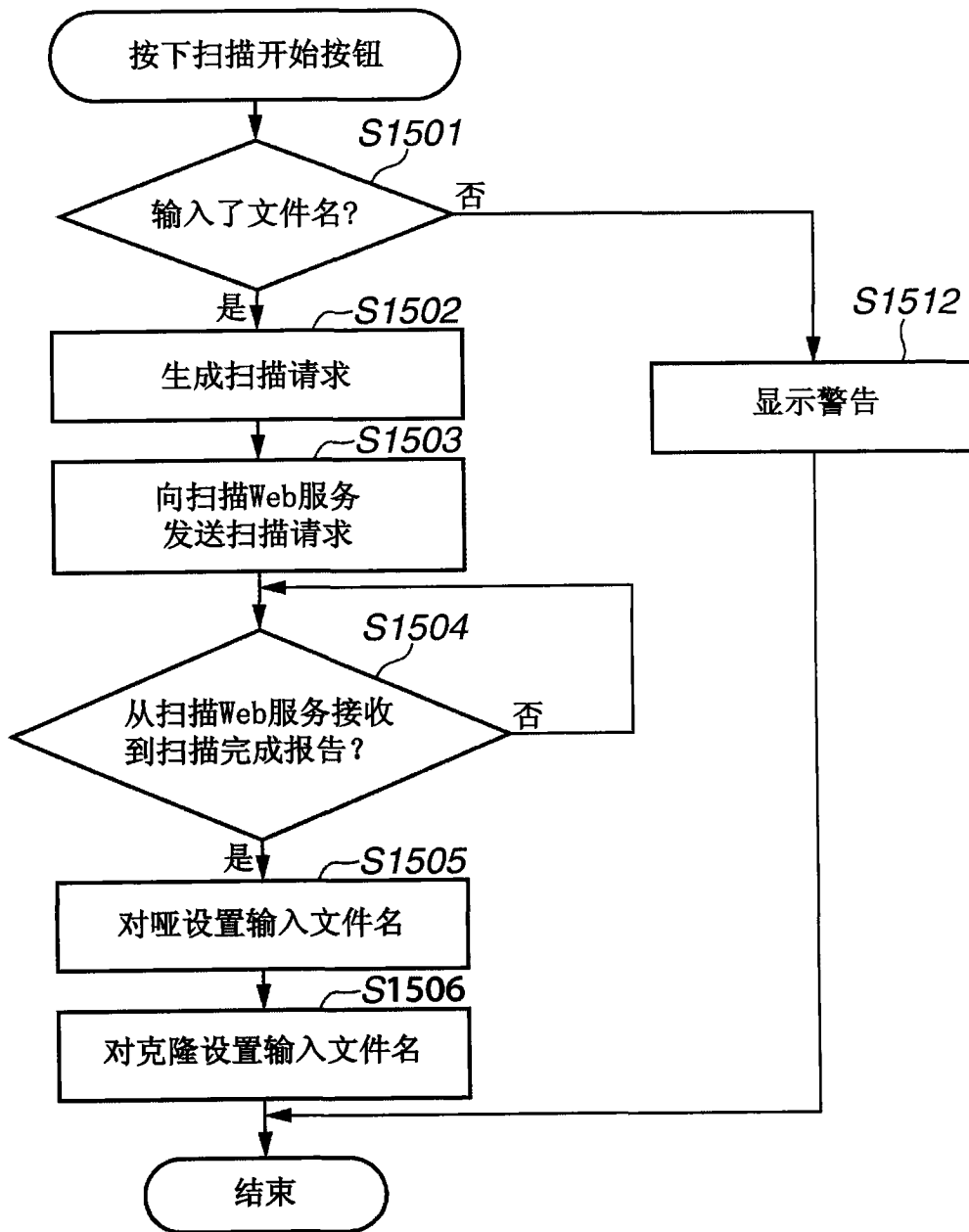


图 15

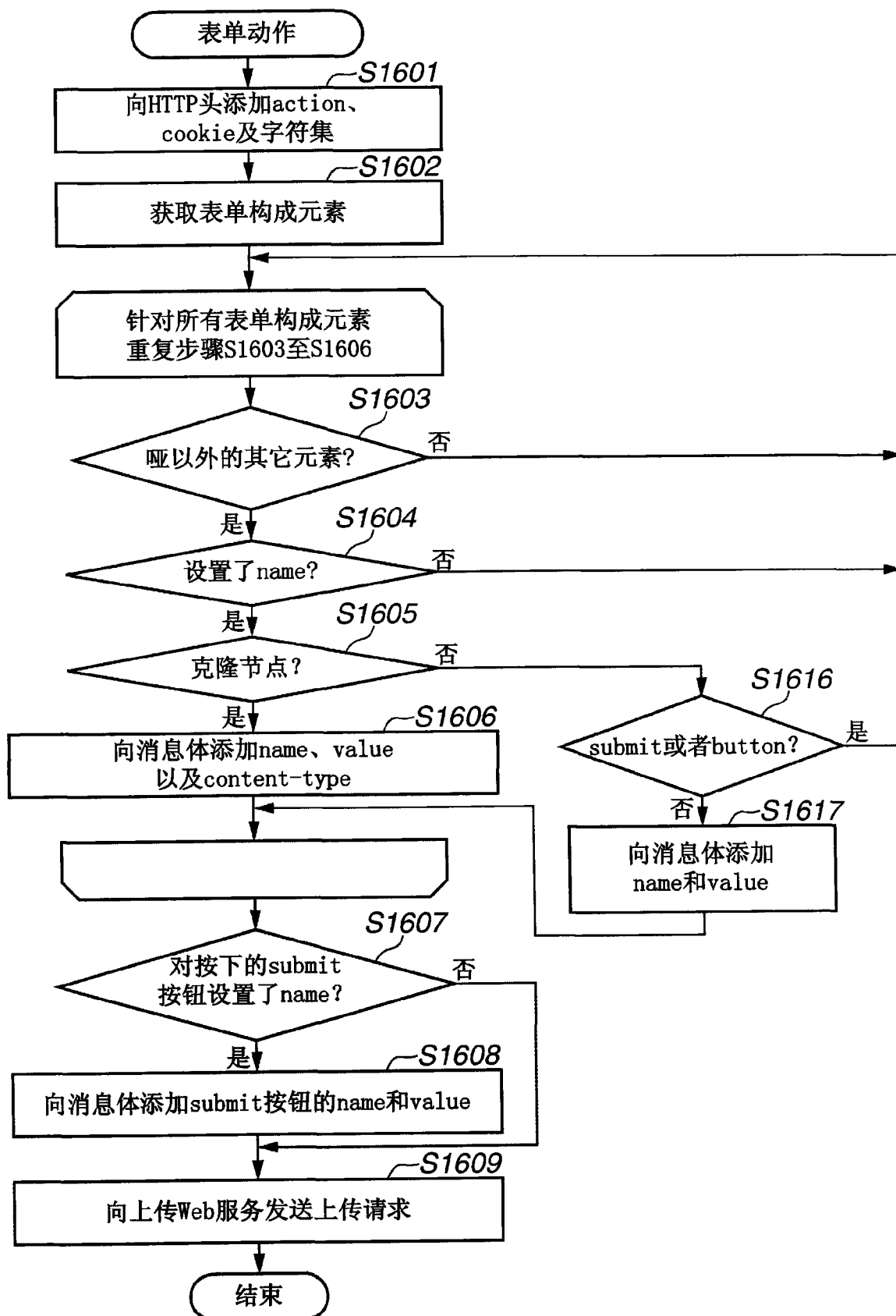


图 16