(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 102246157 B
(45)授权公告日 2013.05.01

(21)申请号 200980150141.9
(22)申请日 2009.11.11
(30)优先权数据
12/330,528 2008.12.09 US
(85)PCT申请进入国家阶段日
2011.06.08
(86)PCT申请的申请数据
PCT/US2009/064034 2009.11.11
(87)PCT申请的公布数据
WO2010/077443 EN 2010.07.08
(73)专利权人 微软公司
地址 美国华盛顿州
(72)发明人 S·乔治 A·卡扎 M·R·哈什
(74)专利代理机构 上海专利商标事务所有限公司 31100
代理人 张欣
(51)Int.Cl.
G06F 17/00(2006.01)
G06F 17/30(2006.01)
(56)对比文件
US 2002/0007393 A1,2002.01.17,
US 2003/0101292 A1,2003.05.29,
US 2008/0209316 A1,2008.08.28,
CN 1475908 A,2004.02.18,
US 2008/0148298 A1,2008.06.19,
US 2008/0184135 A1,2008.07.31,
US 2008/0155554 A1,2008.06.26,
审查员 张文博

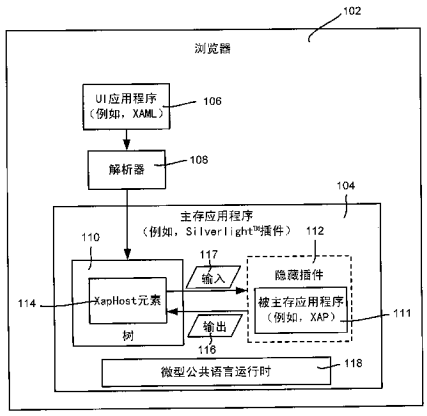
权利要求书2页 说明书6页 附图3页

(54)发明名称

隔离由插件代码主存的应用程序的方法和系统

(57)摘要

本发明描述了一种技术,其中在浏览器中运行的一个应用程序能以防止被主存的应用程序访问浏览器数据或任何其它被主存的应用程序(包括主存)的数据的隔离方式主存另一个应用程序(例如,广告)。主存和/或被主存的应用程序可以是浏览器插件(例如,Microsoft®Silverlight™)应用程序。主存应用程序私下将隐藏插件实例化以禁止隐藏插件访问浏览器数据,并且将被主存的应用程序加载在隐藏插件中。XAML标签元素可用于标识被主存的应用程序以及被主存的应用程序的呈现区域。来自被主存的应用程序的内容在呈现时与来自主存应用程序的内容合成。主存应用程序可向被主存的应用程序提供诸如用于选择相关广告的关键词,和/或可允许被主存的应用程序打开浏览器窗口以显示相关联的网站内容。



CN 102246157 B

1. 在计算环境中，一种用于隔离被主存的应用程序的方法包括，在与浏览器（102）中运行的插件（104）相对应的主存应用程序内主存（202）被主存的应用程序（111），包括防止被主存的应用程序访问所述浏览器的数据或者任何其它被主存的应用程序的数据；

其中，主存被主存的应用程序包括提供用于在与所述主存应用程序相关联的代码内标识所述被主存的应用程序的标签元素，检测所述标签元素，并且作为响应，由所述标签元素实例化隐藏插件，并将所述被主存的应用程序加载到所述隐藏插件中，所述标签元素用作隔离所述隐藏插件的代理。

2. 如权利要求 1 所述的方法，其特征在于，所述隐藏插件对所述浏览器隐藏。

3. 如权利要求 2 所述的方法，其特征在于，防止被主存的应用程序访问数据包括由所述标签元素将所述隐藏插件实例化为禁止其访问浏览器数据或者在所述浏览器中运行的任何其它应用程序的数据。

4. 如权利要求 1 所述的方法，其特征在于，还包括使用所述标签元素将来自所述主存应用程序的输出与来自所述被主存的应用程序的输出合成。

5. 如权利要求 1 所述的方法，其特征在于，还包括使用所述标签元素的一个或多个接口将参数从所述主存应用程序传达到所述被主存的应用程序，或者控制所述被主存的应用程序的动作以尝试控制功耗，或者既将参数从所述主存应用程序传达到所述被主存的应用程序，又控制所述被主存的应用程序的动作以尝试控制功耗。

6. 如权利要求 5 所述的方法，其特征在于，传达所述参数包括提供信息，其中所述被主存的应用程序能通过所述信息选择至少一个相关广告。

7. 如权利要求 1 所述的方法，其特征在于，还包括使用所述标签元素来允许所述被主存的应用程序执行至少一个特权操作。

8. 一种用于隔离被主存的应用程序的系统，包括：

用于运行第一应用程序作为在浏览器中的第一插件的装置，该装置包括用于创建表示所述第一应用程序的元素的元素树的装置；

用于在所述第一插件内主存第二应用程序的装置，该装置包括用于通过在所述元素树中包含用于所述第一插件的主存元素，实例化被禁止访问所述浏览器的数据或所述第一插件的数据的第二插件的装置，以及用于加载第二应用程序以在所述第二插件中运行的装置，其中所述第二应用程序提供用户界面元素，且所述主存元素用作隔离所述第二插件的代理；以及

用于通过处理所述第一应用程序的用户界面元素和所述第二应用程序的用户界面元素来呈现可见输出的装置。

9. 一种用于隔离被主存的应用程序的方法，包括：

运行第一应用程序（104）作为在浏览器（102）中的第一插件，包括创建表示所述第一应用程序的元素的元素树（110）；

在所述第一插件（104）内主存第二应用程序（111），包括通过在所述元素树中包含用于所述第一插件的主存元素（114），实例化（204）被禁止访问所述浏览器的数据或所述第一插件的数据的第二插件，以及加载（206）第二应用程序以在所述第二插件中运行，其中所述第二应用程序提供用户界面元素，且所述主存元素用作隔离所述第二插件的代理；以及

通过处理所述第一应用程序的用户界面元素和所述第二应用程序的用户界面元素来

呈现(210)可见输出。

10. 如权利要求 9 所述的方法，其特征在于，所述第二应用程序对应于广告，并且还包括使用所述主存元素向所述第二应用程序提供与页面内容相对应的一个或多个关键词。

11. 如权利要求 9 所述的方法，其特征在于，还包括使用所述主存元素来允许所述第二应用程序打开浏览器窗口以显示网站内容。

隔离由插件代码主存的应用程序的方法和系统

背景技术

[0001] 当代浏览器允许插件，插件通常包括主存的软件代码，这些主存的软件代码与主存浏览器 / 应用程序交互以提供一些所需功能。一个这样的插件是 Microsoft ® Silverlight™，其提供允许开发和主存 (host) 丰富的 web 应用的平台，其中 web 应用通常包括动画、矢量图形和 / 或媒体（例如，音频 / 视频）内容回放。插件能主存第三方应用程序。

[0002] 通常，这涉及现代应用程序开发的重要功能，即组件化。组件化背后的总体概念是将大的应用程序分解成较小的组件，这些较小的组件执行较大任务中的较小部分、是可重用的并且具有已知的接口。组件化带来的是外部（例如，第二或第三）方为现有应用程序创作组件的能力；例如，对这个概念的一个现代的基于web的具体化是web混搭 (mashup)。

[0003] 然而，当非第一方组件用于诸如由插件主存的应用程序之类的应用程序中时，组件和应用程序之间的信任边界变得重要。组件不应当能够更改用户界面 (UI) 代码的外观和功能，或者监视应用程序数据，等等，除非主存应用程序专门为此而设计且期望如此。

[0004] 概述

[0005] 提供本发明内容以便以简化形式介绍将在以下的详细描述中进一步描述的一些代表性概念。本发明内容不旨在标识出所要求保护的主题的关键特征或必要特征，也不旨在以限制所要求保护的主题的范围的任何方式来使用。

[0006] 简言之，此处所描述的主题的各个方面涉及在浏览器中运行的一个应用程序能主存另一个应用程序的技术，其中上述主存过程是以防止被主存的应用程序访问主存应用程序数据或任何其它被主存的应用程序的数据的方式进行的。在一个方面，主存应用程序是浏览器插件（例如，Microsoft ® Silverlight™）应用程序，浏览器插件应用程序将对浏览器隐藏的另一个插件实例化并且通过将 该另一个插件实例化为禁止其访问浏览器 / 被主存的应用程序数据（例如，元素）或其它应用程序数据来防止被主存的应用程序访问数据。

[0007] 在一个方面，标签元素标识与主存应用程序相关联的代码（例如，XAML）内的主存的应用程序。主存元素被添加到主存应用程序的元素树中以表示被主存的应用程序。标签元素可指定被主存的应用程序的呈现区域。结合（例如，合成的）来自主存应用程序的内容呈现来自被主存的应用程序的内容。

[0008] 在一个方面，被主存的应用程序可以是广告。主存应用程序可向被主存的应用程序提供诸如与页面内容对应的关键词，被主存的应用程序藉此可选择相关广告。被主存的应用程序可被允许执行一些特权操作，诸如允许被主存的应用程序打开浏览器窗口以显示网站内容（例如，当用户点击被主存的应用程序的呈现区域中的广告时调出网站）。

[0009] 结合附图阅读以下具体实施方式，本发明的其他优点会变得显而易见。

附图说明

[0010] 作为示例而非限制，在附图中示出了本发明，附图中相同的附图标记指示相同或相似的元素，附图中：

[0011] 图 1 是示出用于隔离由主存（浏览器插件）应用程序主存的应用程序的示例组件的框图。

[0012] 图 2 是示出隔离被主存的应用程序所采用的示例步骤的流程图。

[0013] 图 3 是示出用于在主存应用程序和被隔离的插件应用程序之间的附加通信的示例组件的框图。

[0014] 图 4 示出可以将本发明的各方面并入其中的计算环境的说明性示例。

具体实施方式

[0015] 此处所描述的技术的各个方面一般涉及隔离模型，其中第一应用程序（例如，浏览器插件等）以隔离的方式主存第二（通常是不可信的、第三方的）应用程序。通常，这通过实例化对 web 浏览器是未知的单独的存储器中 (in-memory) 插件实例以隔离第二应用程序来完成。同时，第二应用程序（例如，广告）在计算机用户看来是作为被主存的应用程序的一部分运行的，因为 例如它的可见输出是与主存应用程序的可见输出一起呈现的。然而，可以理解，被主存的应用程序不能访问与主存应用程序或 web 浏览器相关的信息，除非主存应用程序希望提供某些信息。

[0016] 尽管使用 Microsoft ® Silverlight™（跨平台、跨浏览器插件）作为主存应用程序的示例，应当理解此处所述的任何示例都是非限制性的示例。如此，本发明不限于此处所描述的任何特定实施例、方面、概念、结构、功能或示例。相反，此处所描述的实施例、方面、概念、结构、功能或示例中的任一个都是非限制性的，并且本发明一般能够以在计算和主存应用程序方面提供好处和优点的各种方式来使用。

[0017] 图 1 示出与此处所描述的应用程序隔离相关的各个方面。通常，利用诸如 Microsoft ® Silverlight™ 插件之类的主存应用程序 104 来加载浏览器 102。在一个实现中，主存应用程序 104 对应于诸如用 XAML（可扩展应用程序标记语言）编写的用户界面应用程序代码 106，该用户界面应用程序代码 106 被解析器 108 解析成与树 110 相对应的各种元素。如所知道的，这些元素被处理以在浏览器页面上的可见输出中呈现。

[0018] 在一个实现中，诸如第三方广告之类的应用程序 111 由主存应用程序 104 主存为隐藏（对浏览器而言是单独的，不可发现的）插件 112 的一部分，使得将隐藏插件 112 与浏览器 102 以及与主存应用程序的数据在程序上隔离，但是允许被主存的应用程序 111 参与主存应用程序的呈现、输入和布局。呈现、输入和布局由隐藏插件 112 管理，从而例如可见输出限于为被主存的应用程序的输出所保留的用户界面区域。注意，隐藏插件 112 可以是另一个 Silverlight™ 插件，即由主存 Silverlight™ 插件私下创建（对浏览器是未知的）的另一个实例。此外，注意可使用其它类型的插件，只要它们能被实例化或者以其它方式配置为防止它们与 HTML 浏览器发生任何直接通信，例如将插件实例化以使其不能访问浏览器文档对象模型 (DOM)。

[0019] 为此，在一个示例实现中，（例如，在 XAML 应用程序代码 106 中）提供了标识被主存的应用程序的源且表示隔离边界的标签元素，在该隔离边界中将隐藏插件 112 实例化以用于加载被主存的（隔离的）应用程序 111：

[0020] <Canvas><！—主存应用程序的一部分 -->

[0021] <XapHost Source =”http://www.advertisements.com/ad.xap”Height

＝”300”Width ＝”350”/><！—隔离边界 -->

[0022] </Canvas>

[0023] 标签元素在图 1 中由 XapHost 元素 114 来表示。除其它操作以外，XapHost 元素 114 负责在存储器中在浏览器 102 不可发现的位置将隐藏插件 112 实例化和初始化；这可经由专用 API 等来完成。被主存的应用程序 111 除非经由用作隔离隐藏插件 112 的代理的 XapHost 元素 114 才能参与主存应用程序 104 的树 110。例如，被主存的应用程序 111 除了遍历至它自己的最顶层节点以外不能遍历树，其最顶层节点对应于它的应用程序的根节点。注意，它不能访问驻留在主存应用程序树中的 XapHost 元素。相反，被主存的应用程序 111 被限于仅仅经由 XapHost 元素 114 提供输出 116 用于呈现，（尽管 XapHost 元素 114 可提供诸如初始化参数之类的附加数据和 / 或允许对被主存的应用程序 111 的某些特权，如以下参考图 3 所述）。从 XapHost 元素 114 传达到隐藏插件 112/ 被主存的应用程序 111 的这些数据以及任何其它数据在图 1 中被表示为输入 117。

[0024] 如经由图 2 的示例步骤所一般地表示的，当 XAML 应用程序代码 106 被解析时或者当 XapHost 元素 114 被程序地添加到树 110 时（步骤 202），在步骤 204 和 206 处 XapHost 元素 114 初始化隐藏插件 112 并且加载由“源 (Source)”属性所指定的相应的被主存的应用程序 111（例如，XAP），（其在上述示例 XAML 语言中被标识为“http://www.advertisements.com/ad.xap”）。被主存的应用程序 111 不能访问浏览器的 HTML 域对象模型（例如，启用 Html 访问 (EnableHtmlAccess) 特性被设置为错误）并且被加载在它自己的应用域中，例如，（公共语言运行时，或 CLR 应用域），它自己可访问全局静态变量等；注意，Silverlight™ 包含它自己的微型 CLR 118。在没有域对象模型 (DomBridge) 访问的情况下，被主存的应用程序 111 不能访问页面上的任何其它内容，包括页面自身，或者其它插件。

[0025] 同样如在图 2 中经由步骤 208 和 210 所示，除了实例化和初始化隐藏插件 112 且加载应用程序 111 以外，XapHost 元素 114 将输入和布局通知从主存应用程序 104 引导至被主存的应用程序 111。此外，XapHost 元素 114 处理被主存的应用程序 111 的输出，使得该输出能由主存应用程序的插件来合成；（注 意，诸如音频和 / 或触觉输出之类的任何其它输出可被类似地混合）。步骤 212 表示在适当时间拆卸被主存的应用程序，例如，在关闭之后或者当 XapHost 元素 114 上的相应的引用计数变为零时（例如，当用户界面的该部分不再被呈现时）。

[0026] 在一个方面，通常在图 3 中表示的，XapHost 元素 114 可包括一个或多个接口（例如，API 330（多个），以及上述其它专用 API），这些接口允许主存应用程序 104 与被主存的应用程序 111 通信，例如传递初始化参数 332。例如，在广告情形中，初始化参数可包括在主存页面上呈现的关键词的集合，由此广告应用程序可服务与这些关键词中的一个或多个相对应的相关广告。

[0027] 在同样通常在图 3 中表示的另一个方面，经由 API 330，XapHost 元素 114 可允许被主存的应用程序执行某些特权操作，（或者根据适当的调用 334 代表其自身执行操作）。一个示例是响应于用户发起的动作打开新的浏览器窗口，诸如调出与被点击广告相对应的网站。

[0028] 另一个优点是控制功耗的能力。例如，在现代计算机使用中广告可能是较大的功耗源。利用插件主存（例如，XapHost）模型，降低的功率模式可诸如通过禁用动画以及其

它功率消耗操作来节省功率。

[0029] 示例性操作环境

[0030] 图 4 示出了其上可实现图 1-3 的示例的合适的计算和联网环境 400 的示例。计算系统环境 400 只是合适计算环境的一个示例，而非意在暗示对本发明使用范围或功能有任何限制。也不应该将计算环境 400 解释为对示例性操作环境 400 中示出的任一组件或其组合有任何依赖性或要求。

[0031] 本发明可用各种其他通用或专用计算系统环境或配置来操作。适用于本发明的公知计算系统、环境、和 / 或配置的示例包括但不限于 :个人计算机、服务器计算机、手持式或膝上型设备、平板设备、多处理器系统、基于微处理器的系统、机顶盒、可编程消费电子产品、网络 PC、微型计算机、大型计算机、包括任何以上系统或设备的分布式计算环境等等。

[0032] 本发明可在诸如程序模块等由计算机执行的计算机可执行指令的通用上下文中描述。一般而言，程序模块包括执行特定任务或实现特定抽象数据类型的例程、程序、对象、组件、数据结构等等。本发明也可以在其中任务由通过 通信网络链接的远程处理设备执行的分布式计算环境中实现。在分布式计算环境中，程序模块可以位于包括存储器存储设备在内的本地和 / 或远程计算机存储介质中。

[0033] 参考图 4，用于实现本发明的各方面的示例性系统可包括计算机 410 形式的通用计算设备。计算机 410 的组件可以包括但不限于 :处理单元 420、系统存储器 430 和将包括系统存储器在内的各种系统组件耦合至处理单元 420 的系统总线 421。系统总线 421 可以是若干类型的总线结构中的任一种，包括使用各种总线体系结构中的任一种的存储器总线或存储器控制器、外围总线，以及局部总线。作为示例而非限制，这样的架构包括工业标准架构 (ISA) 总线、微通道架构 (MCA) 总线、增强型 ISA(EISA) 总线、视频电子标准协会 (VESA) 局部总线、以及也称为夹层 (Mezzanine) 总线的外围组件互连 (PCI) 总线。

[0034] 计算机 410 通常包括各种计算机可读介质。计算机可读介质可以是能由计算机 410 访问的任何可用介质，并包含易失性和非易失性介质以及可移动、不可移动介质。作为示例而非限制，计算机可读介质可以包括计算机存储介质和通信介质。计算机存储介质包括以存储诸如计算机可读的指令、数据结构、程序模块或其他数据之类的信息的任何方法或技术实现的易失性和非易失性、可移动和不可移动介质。计算机存储介质包括，但不仅限于，RAM、ROM、EEPROM、闪存或其他存储器技术、CD-ROM、数字多功能盘 (DVD) 或其他光盘存储、磁带盒、磁带、磁盘存储或其他磁存储设备，或可以用来存储所需信息并可以被计算机 410 访问的任何其他介质。通信介质通常以诸如载波或其他传输机制的已调制数据信号来体现计算机可读指令、数据结构、程序模块或其他数据，并包括任意信息传送介质。术语“已调制数据信号”指的是一个或多个特征以在信号中编码信息的方式被设定或更改的信号。作为示例而非限制，通信介质包括有线介质，如有线网络或直接线连接，以及如声学、RF、红外及其他无线介质之类的无线介质。上面各项中的任何项的组合也包括在计算机可读介质的范围内。

[0035] 系统存储器 430 包括易失性和 / 或非易失性存储器形式的计算机存储介质，如只读存储器 (ROM)431 和随机存取存储器 (RAM)432。基本输入 / 输出系统 433(BIOS) 包括如在启动时帮助在计算机 410 内的元件之间传输信息的基 本例程，它通常储存在 ROM 431 中。RAM 432 通常包含处理单元 420 可以立即访问和 / 或目前正在操作的数据和 / 或程序模块。

作为示例而非限制，图 4 示出了操作系统 434、应用程序 435、其他程序模块 436 和程序数据 437。

[0036] 计算机 410 还可以包括其他可移动 / 不可移动、易失性 / 非易失性计算机存储介质。仅作为示例，图 4 示出了从不可移动、非易失性磁介质中读取或向其写入的硬盘驱动器 441，从可移动、非易失性磁盘 452 中读取或向其写入的磁盘驱动器 451，以及从诸如 CD ROM 或其他光学介质等可移动、非易失性光盘 456 中读取或向其写入的光盘驱动器 455。可以在示例性操作环境中使用的其他可移动 / 不可移动、易失性 / 非易失性计算机存储介质包括但不限于，磁带盒、闪存卡、数字多功能盘、数字录像带、固态 RAM、固态 ROM 等等。硬盘驱动器 441 通常由不可移动存储器接口，诸如接口 440 连接至系统总线 421，磁盘驱动器 451 和光盘驱动器 455 通常由可移动存储器接口，诸如接口 450 连接至系统总线 421。

[0037] 以上描述并在图 4 中示出的驱动器及其相关联的计算机存储介质为计算机 410 提供了对计算机可读指令、数据结构、程序模块和其他数据的存储。例如，在图 4 中，硬盘驱动器 441 被示为存储操作系统 444、应用程序 445、其他程序模块 446 和程序数据 447。注意，这些组件可以与操作系统 434、应用程序 435、其他程序模块 436 和程序数据 437 相同，也可以与它们不同。操作系统 444、应用程序 445、其他程序模块 446 和程序数据 447 在这里被标注了不同的附图标记是为了说明至少它们是不同的副本。用户可通过诸如平板或电子数字化仪 464、话筒 463、键盘 462 和定点设备 461（通常指的是鼠标、跟踪球或触摸垫）等输入设备向计算机 410 输入命令和信息。图 4 中未示出的其他输入设备可以包括操纵杆、游戏手柄、圆盘式卫星天线、扫描仪等。这些和其他输入设备通常通过耦合至系统总线的用户输入接口 460 连接至处理单元 420，但也可以由其他接口和总线结构，诸如并行端口、游戏端口或通用串行总线 (USB) 来连接。监视器 491 或其他类型的显示设备也通过接口，诸如视频接口 490，连接至系统总线 421。监视器 491 也可以与触摸屏面板等集成。注意，监视器和 / 或触摸屏面板可以在物理上耦合至其中包括计算设备 410 的外壳，诸如在平板型个人计算机中。此外，诸如计算设备 410 等计算机还可以包括其他外围输 出设备，诸如扬声器 495 和打印机 496，它们可以通过输出外围接口 494 等连接。

[0038] 计算机 410 可以使用到一个或多个远程计算机（如远程计算机 480）的逻辑连接来在联网环境中操作。远程计算机 480 可以是个人计算机、服务器、路由器、网络 PC、对等设备或其他常见网络节点，并且通常包括许多或所有以上关于计算机 410 所描述的元件，尽管在图 4 中仅示出了存储器存储设备 481。图 4 中所示的逻辑连接包括一个或多个局域网 (LAN) 471 和一个或多个广域网 (WAN) 473，但也可以包括其他网络。这样的联网环境在办公室、企业范围计算机网络、内联网和因特网中是常见的。

[0039] 当在 LAN 联网环境中使用时，计算机 410 通过网络接口或适配器 471 连接至 LAN 470。当在 WAN 联网环境中使用时，计算机 410 通常包括调制解调器 472 或用于通过诸如因特网等 WAN 473 建立通信的其他装置。可为内置或可为外置的调制解调器 472 可以经由用户输入接口 460 或其他合适的机制连接至系统总线 421。诸如包括接口和天线的无线联网组件 474 可以通过诸如接入点或对等计算机等合适的设备耦合到 WAN 或 LAN。在联网环境中，参考计算机 410 所描述的程序模块，或其部分，可以存储在远程存储器存储设备中。作为示例而非限制，图 4 示出远程应用程序 485 驻留在存储器设备 481 上。可以理解，所示的网络连接是示例性的，也可以使用在计算机之间建立通信链路的其他手段。

[0040] 辅助子系统 499（例如，用于内容的辅助显示）可经由用户接口 460 连接，从而即使计算机系统的主要部分处于低功率状态中，也允许诸如程序内容、系统状态和事件通知等数据被提供给用户。辅助子系统 499 可连接至调制解调器 472 和 / 或网络接口 470，从而在主处理单元 420 处于低功率状态中时，也允许在这些系统之间进行通信。

[0041] 结论

[0042] 尽管本发明易于作出各种修改和替换构造，但其某些说明性实施例在附图中示出并在上面被详细地描述。然而应当了解，这不旨在将本发明限于所公开的具体形式，而是相反地，旨在覆盖落入本发明的精神和范围之内的所有修改、替换构造和等效方案。

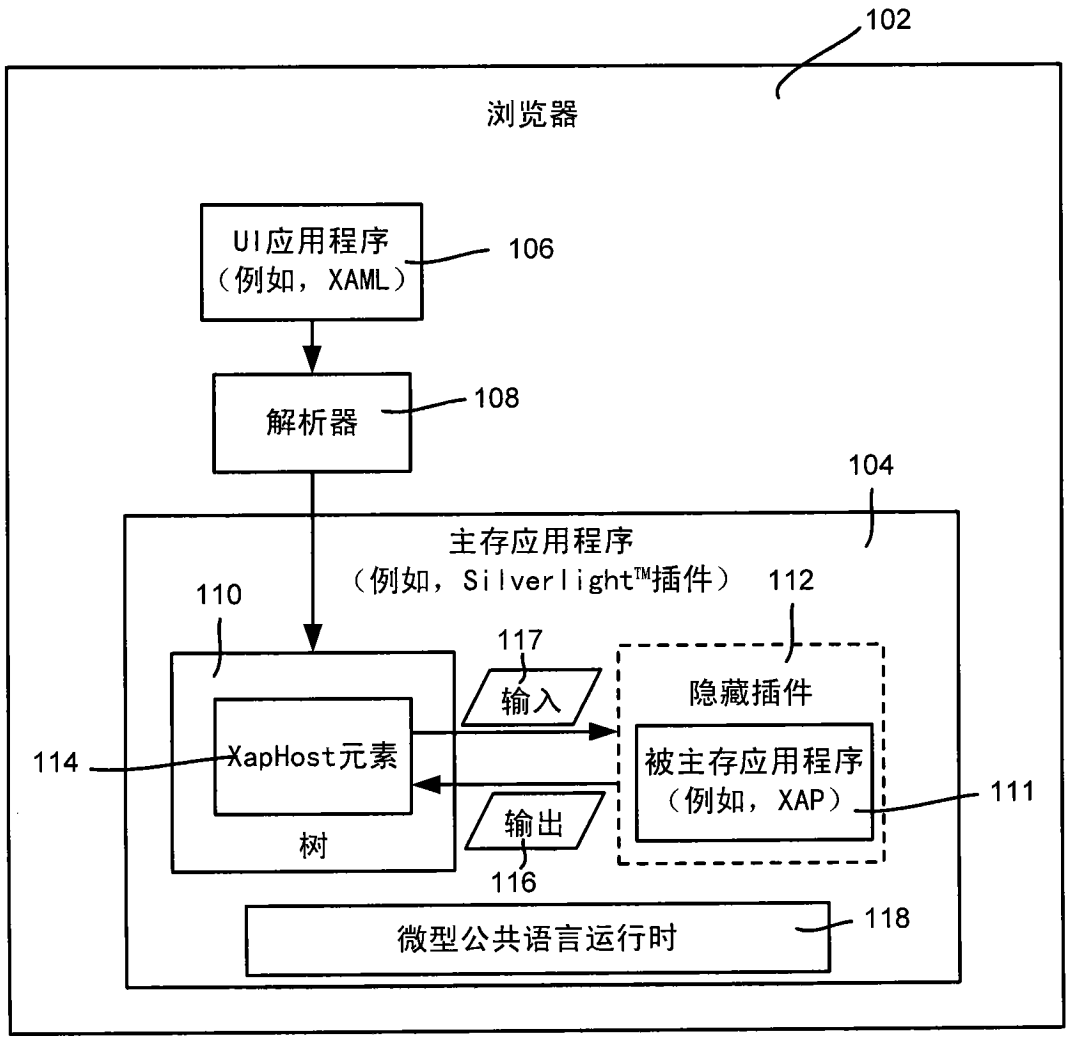

102
浏览器
UI应用程序
（例如，XAML）
106
解析器
108
104
主存应用程序
（例如，Silverlight™插件）
112
110
117
输入
隐藏插件
114
XapHost元素
被主存应用程序
（例如，XAP）
111
输出
树
116
微型公共语言运行时
118

图 1

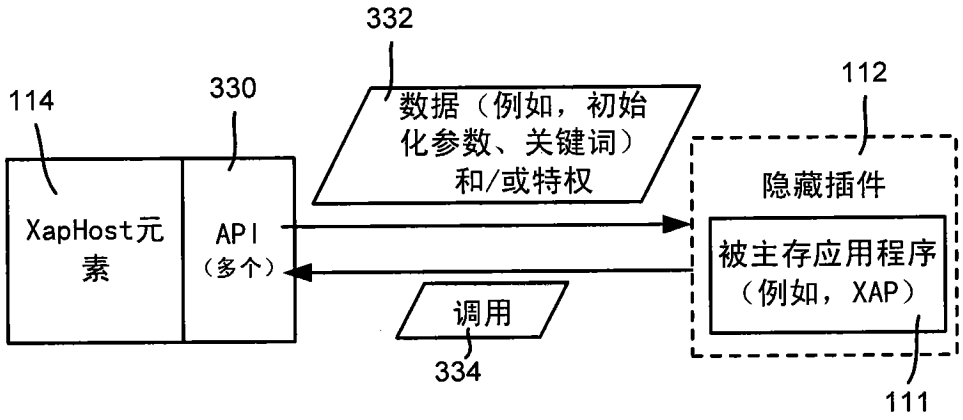

332
112
114
330
数据（例如，初始
化参数、关键词）
和/或特权
隐藏插件
XapHost元
素
API
（多个）
被主存应用程序
（例如，XAP）
调用
334
111

图 3

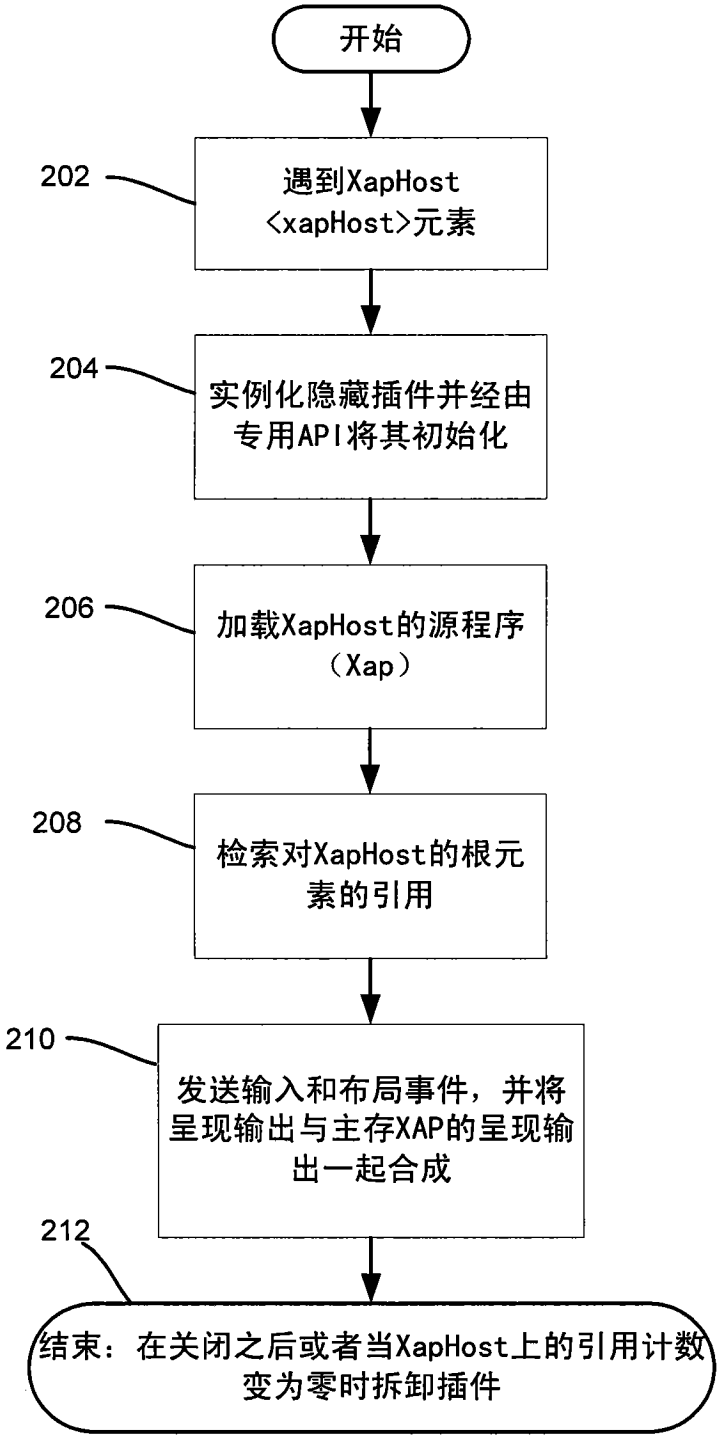

开始
202
遇到XapHost
<xapHost>元素
204
实例化隐藏插件并经由
专用API将其初始化
206
加载XapHost的源程序
（Xap）
208
检索对XapHost的根元
素的引用
210
发送输入和布局事件，并将
呈现输出与主存XAP的呈现输
出一起合成
212
结束：在关闭之后或者当XapHost上的引用计数
变为零时拆卸插件

图 2

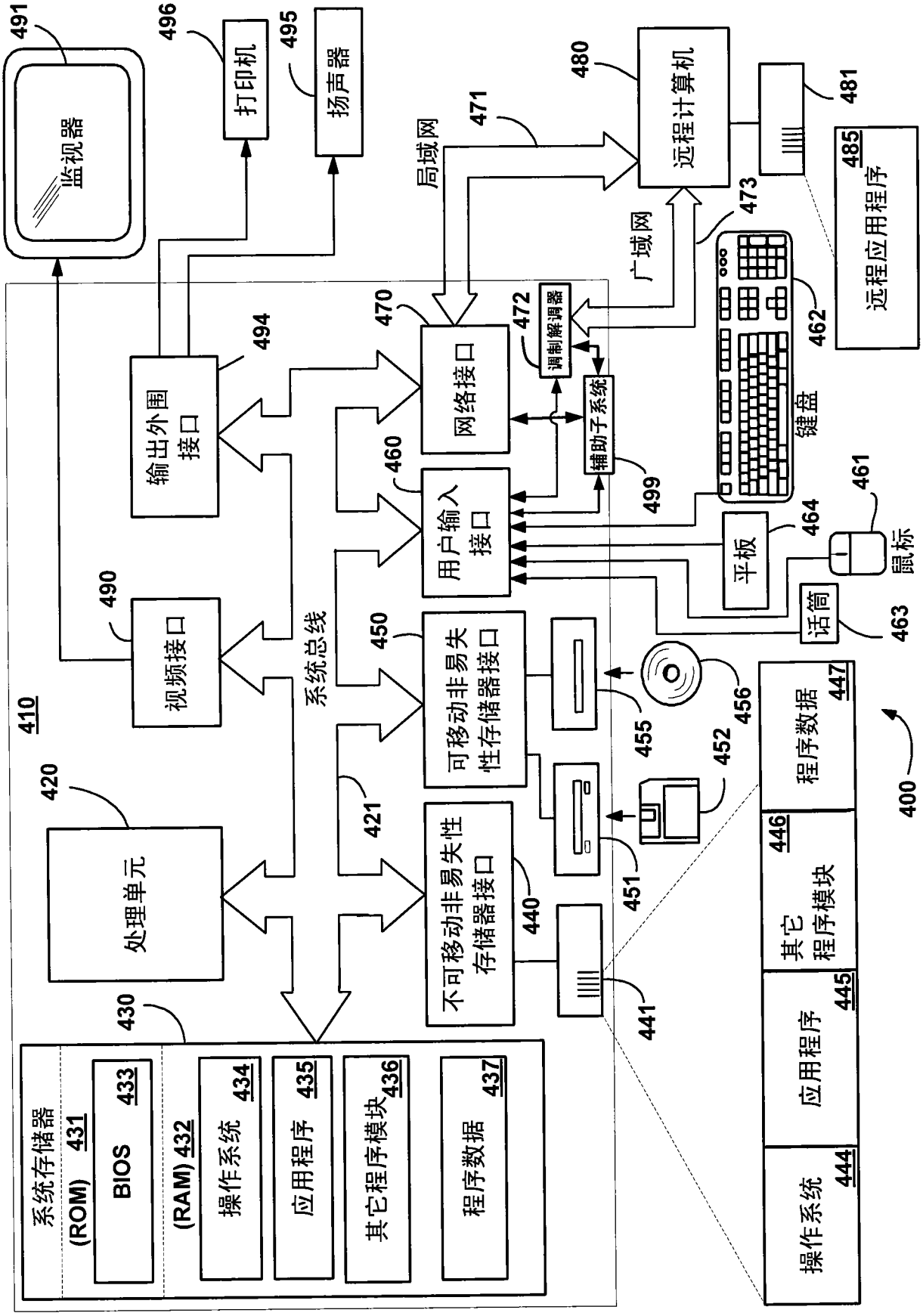

491
监视器
496
打印机
495
扬声器
480
远程计算机
481
485
远程应用程序
局域网
471
广域网
473
462
键盘
494
输出外围接口
470
网络接口
472
调制解调器
499
辅助子系统
460
用户输入接口
464
平板
461
鼠标
463
话筒
490
视频接口
450
可移动非易失性存储器接口
456
455
452
451
441
440
不可移动非易失性存储器接口
421
420
处理单元
410
系统总线
430
系统存储器
(ROM) 431
BIOS 433
(RAM) 432
操作系统 434
应用程序 435
其它程序模块 436
程序数据 437
操作系统 444
应用程序 445
其它程序模块 446
程序数据 447
400

图 4