

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.

G06F 17/30 (2006.01)

G06F 9/44 (2006.01)

H04N 1/00 (2006.01)



[12] 发明专利说明书

专利号 ZL 200510104040.1

[45] 授权公告日 2009年5月6日

[11] 授权公告号 CN 100485673C

[22] 申请日 2005.9.15

[21] 申请号 200510104040.1

[30] 优先权

[32] 2004.9.15 [33] JP [31] 2004-268483

[73] 专利权人 佳能株式会社

地址 日本东京

[72] 发明人 山本雅仁

[56] 参考文献

CN1482562A 2004.3.17

EP1030276A2 2000.8.23

WO2004/031984A1 2004.4.15

Access Control of XML Documents Considering UpdateOperations. LIM C. H ET AL. PROCEEDINGS OF THE ACM WORKSHOP ON XML SECURITY 2003. 2003

An Embedded Web Server Architecture for XML - Based Network Management. JU H. T ET AL. IEEE/IFIP NETWORK OPERATIONS AND MANAGEMENT SYMPOSIUM. 2002

审查员 韩燕_2

[74] 专利代理机构 中国国际贸易促进委员会专利商标事务所

代理人 李德山

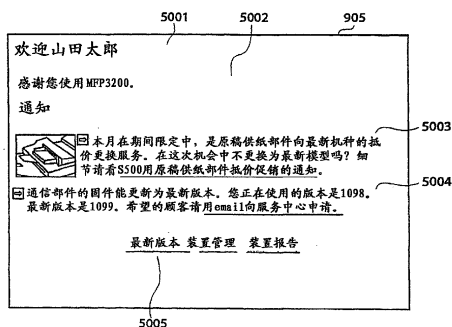
权利要求书 7 页 说明书 63 页 附图 45 页

[54] 发明名称

嵌入装置及其控制方法

[57] 摘要

在提供嵌入装置的图像处理装置(110)中, Web 浏览器模块(211)至少加载包含由标记语言记述的文书数据即在其中嵌入脚本或参照脚本的信息的文书数据的 Web 内容数据, 根据该加载的 Web 内容数据进行显示。控制部件(111)控制该图像处理装置的信息和动作。脚本解释程序(805)嵌入所述 Web 浏览器模块中, 解释所述加载的 Web 内容数据中包含的文书数据, 处理该文书数据中嵌入的脚本或由参照信息参照的脚本。控制 API 模块(218)提供所述 Web 浏览器模块用于根据所述脚本的处理来访问所述 Web 浏览器模块内管理的 DOM 的借口。控制 API 模块提供所述脚本解释程序用于根据所述脚本的处理来访问由所述控制部件控制的该图像处理装置的信息和动作的接口。



1. 一种嵌入装置，其特征在于，包括：

加载至少包含由标记语言记述的文书数据即其中嵌入脚本或参照脚本的信息的文书数据的内容数据，并根据该加载的内容数据来进行显示的浏览器部件；

控制该嵌入装置的信息和动作的控制部件；

解释所述加载的内容数据中包含的文书数据，并处理嵌入该文书数据中的脚本或由参照信息参照的脚本的脚本处理部件，即嵌入所述浏览器部件中的脚本处理部件；

提供用于所述脚本处理部件根据所述脚本的处理来访问由所述控制部件控制的该嵌入装置的信息和动作的接口的第一接口部件；和

提供用于所述脚本处理部件根据所述脚本的处理来访问在所述文书数据的内部数据构造的接口的第二接口部件，

所述脚本处理部件把根据通过所述第一接口部件访问由所述控制部件控制的该嵌入装置的信息和动作而取得的结果所生成的一部分文书数据插入所述文书数据中的与通过所述第二接口部件访问所述文书数据的内部数据构造而取得的结果对应的位置。

2. 根据权利要求1所述的嵌入装置，其特征在于，

所述第二接口部件构筑用于所述脚本处理部件访问所述文书数据的内部数据构造的文档对象模型，把所述文书数据的各要素分别与所述脚本处理部件作为处理对象的各对象关联，把所述文书数据的各要素的属性与该要素所对应的对象的属性关联，把对于所述文书数据的各要素的操作与对于该要素所对应的对象的操作关联，把所述文书数据的各要素之间的关系与对应于该要素间的对象间的关系关联，把对于所述文书数据或其要素生成的事件与对于该文书数据或要素所对应的对象生成的事件关联，来构筑所述文档对象模型。

3. 根据权利要求2所述的嵌入装置，其特征在于，

所述脚本处理部件与对于所述文书数据或其要素生成的事件呼

应，通过所述第一接口部件访问由所述控制部件控制的该嵌入装置的信息和动作。

4. 根据权利要求1所述的嵌入装置，其特征在于，

所述第一接口部件构筑用于所述脚本处理部件访问由所述控制部件控制的该嵌入装置的信息和动作的装置对象模型，把构成该嵌入装置的各种资源分别与所述脚本处理部件作为处理对象的各对象关联，把所述各资源的属性与该资源所对应的对象的属性关联，把对于所述各资源的操作与该资源所对应的对象的操作关联，把所述各资源间的关系与该资源间所对应的对象间的关系关联，来构筑装置对象模型。

5. 根据权利要求4所述的嵌入装置，其特征在于，

在所述装置对象模型中，继承抽象-具体关系，具体类具有抽象类的属性和操作中至少一方的至少一部分，属于各类的对象具有该类的属性和操作中至少一方的至少一部分。

6. 根据权利要求4所述的嵌入装置，其特征在于，

在所述装置对象模型中，表现全体-部分的聚合关系，全体对象具有用于探索与该全体对象关联的0个以上的部分对象的操作或属性。

7. 根据权利要求4所述的嵌入装置，其特征在于，

进而把对于该嵌入装置或其资源生成的事件和例外分别与对于该嵌入装置或其资源所对应的对象生成的事件和例外关联，来构筑所述装置对象模型。

8. 根据权利要求1所述的嵌入装置，其特征在于，

当由用户指示把所述脚本处理部件通过所述第一接口部件访问由所述控制部件控制的该嵌入装置的信息和动作而取得的结果向该嵌入装置的外部输出时，所述浏览器部件禁止与该指示对应的外部输出。

9. 根据权利要求1所述的嵌入装置，其特征在于，

当由用户指示把所述脚本处理部件通过所述第一接口部件访问由所述控制部件控制的该嵌入装置的信息和动作而取得的结果向该嵌

入装置的外部输出时，所述浏览器部件预先或在指示时对用户确认可否进行与该指示对应的外部输出。

10. 根据权利要求4所述的嵌入装置，其特征在于，

按照构成所述装置对象模型的各对象保持访问权限，所述脚本处理部件尝试对没有访问权限的对象、对象的属性或对象的操作进行访问时，所述第一接口部件禁止该访问。

11. 根据权利要求10所述的嵌入装置，其特征在于，

还具有：设定对于构成所述装置对象模型的各对象的访问权限的设定部件。

12. 根据权利要求10所述的嵌入装置，其特征在于，

构成所述装置对象模型的各对象继承该对象所属的类中保持的访问权限。

13. 根据权利要求10所述的嵌入装置，其特征在于，

作为构成装置对象模型的各对象的访问权限，设定与该对象存在聚合关系，并且在包含的对象中保持的访问权限。

14. 一种嵌入装置，其特征在于，包括：

加载至少包含由标记语言记述的文书数据的内容数据，并根据该加载的内容数据来进行显示的浏览器部件；

控制该嵌入装置的信息和动作的控制部件；

提供所述浏览器部件用于访问由所述控制部件控制的该嵌入装置的信息和动作的接口的接口部件；

所述浏览器部件处理嵌入到上述文书数据中的脚本或者由嵌入到上述文书数据中的参照信息所参照的脚本，根据该处理通过所述接口部件访问由所述控制部件控制的该嵌入装置的信息和动作，根据通过该访问而取得的结果，识别正在使用该嵌入装置的用户，显示与该用户对应的文书的内容。

15. 一种嵌入装置，其特征在于，包括：

加载至少包含由标记语言记述的文书数据的内容数据，并根据该加载的内容数据来进行显示的浏览器部件；

控制该嵌入装置的信息和动作的控制部件；

提供所述浏览器部件用于访问由所述控制部件控制的该嵌入装置的信息和动作的接口的接口部件；

所述浏览器部件处理嵌入到上述文书数据中的脚本或者由嵌入到上述文书数据中的参照信息所参照的脚本，根据该处理通过所述接口部件访问由所述控制部件控制的该嵌入装置的信息和动作，根据通过该访问而取得的结果，来判别该嵌入装置的固有信息或该嵌入装置的状态，显示与该判别结果对应的文书的内容。

16. 根据权利要求 15 所述的嵌入装置，其特征在于，

所述固有信息包含该嵌入装置的种类、安装的选项的种类、该嵌入装置以及选项部件的各版本中至少任意一个。

17. 根据权利要求 15 所述的嵌入装置，其特征在于，

所述文书包含关于该嵌入装置以及该嵌入装置中安装的选项部件的至少一个的手册、对用户的提议、参照所述手册或所述提议的信息的至少一个。

18. 一种嵌入装置，其特征在于，包括：

加载至少包含由标记语言记述的文书数据的内容数据，并根据该加载的内容数据来进行显示的浏览器部件；

控制该嵌入装置的信息和动作的控制部件；

提供所述浏览器部件用于访问由所述控制部件控制的该嵌入装置的信息和动作的接口的接口部件；

所述浏览器部件处理嵌入到上述文书数据中的脚本或者由嵌入到上述文书数据中的参照信息所参照的脚本，根据该处理通过所述接口部件访问由所述控制部件控制的该嵌入装置的信息和动作，根据通过该访问而取得的结果，来判别该嵌入装置中是否存在内部程序，如果存在内部程序，就显示包含用于指示该内部程序的起动的用户输入要素的文书。

19. 一种嵌入装置，其特征在于，包括：

控制该嵌入装置的信息和动作的控制部件；

处理脚本的脚本处理部件；

提供所述脚本处理部件根据所述脚本的处理用于访问由所述控制部件控制的该嵌入装置的信息和动作的接口的接口部件；

所述接口部件构筑用于所述脚本处理部件访问由所述控制部件控制的该嵌入装置的信息和动作的装置对象模型，通过把构成该嵌入装置的各种资源分别与所述脚本处理部件作为处理对象的各对象关联，把所述各资源的属性与该资源所对应的对象的属性关联，把对于所述各资源的操作与该资源所对应的对象的操作关联，把所述各资源间的关系与该资源间所对应的对象间的关系关联，来构筑所述装置对象模型。

20. 根据权利要求 19 所述的嵌入装置，其特征在于，

在所述装置对象模型中，继承抽象-具体关系，具体类具有抽象类的属性和操作中至少一方的至少一部分，属于各类的对象具有该类的属性和操作中至少一方的至少一部分。

21. 根据权利要求 19 所述的嵌入装置，其特征在于，

在所述装置对象模型中，表现全体-部分的聚合关系，全体对象具有用于探索与该全体对象关联的 0 个以上的部分对象的操作或属性。

22. 根据权利要求 19 所述的嵌入装置，其特征在于，

把对于该嵌入装置或其资源生成的事件和例外分别与对于该嵌入装置或其资源所对应的对象生成的事件和例外关联，来构筑所述装置对象模型。

23. 一种嵌入装置的控制方法，其特征在于，包括：

加载至少包含由标记语言记述的文书数据即其中嵌入脚本或参照脚本的信息的文书数据的内容数据，根据该加载的内容数据来进行显示的浏览器步骤；

控制该嵌入装置的信息和动作的控制步骤；

解释所述加载的内容数据中包含的文书数据，并处理嵌入该文书数据中的脚本或由参照信息参照的脚本的脚本处理步骤，即嵌入所述

浏览器步骤中的脚本处理步骤;

提供用于所述脚本处理步骤根据所述脚本的处理来访问由所述控制步骤控制的该嵌入装置的信息和动作的接口的第一接口步骤; 和

提供用于在所述脚本处理步骤中的根据所述脚本的处理来访问所述文书数据的内部数据构造的接口的第二接口步骤,

所述脚本处理步骤把根据通过所述第一接口步骤访问由所述控制步骤控制的该嵌入装置的信息和动作而取得的结果所生成的一部分文书数据插入所述文书数据中的与通过所述第二接口步骤访问所述文书数据的内部数据构造而取得的结果对应的位置。

24. 一种嵌入装置的控制方法, 其特征在于, 包括:

加载至少包含由标记语言记述的文书数据的内容数据, 根据该加载的内容数据来进行显示的浏览器步骤;

控制该嵌入装置的信息和动作的控制步骤;

提供所述浏览器步骤用于访问由所述控制步骤控制的该嵌入装置的信息和动作的接口的接口步骤;

所述浏览器步骤处理嵌入到上述文书数据中的脚本或者由嵌入到上述文书数据中的参照信息所参照的脚本, 根据该处理通过所述接口步骤访问由所述控制步骤控制的该嵌入装置的信息和动作, 根据通过该访问而取得的结果, 识别正在使用该嵌入装置的用户, 显示与该用户对应的文书的内容。

25. 一种嵌入装置的控制方法, 其特征在于, 包括:

加载至少包含由标记语言记述的文书数据的内容数据, 根据该加载的内容数据来进行显示的浏览器步骤;

控制该嵌入装置的信息和动作的控制步骤;

提供所述浏览器步骤用于访问由所述控制步骤控制的该嵌入装置的信息和动作的接口的接口步骤;

所述浏览器步骤处理嵌入到上述文书数据中的脚本或者由嵌入到上述文书数据中的参照信息所参照的脚本, 根据该处理通过所述接口步骤访问由所述控制步骤控制的该嵌入装置的信息和动作, 根据通

过该访问而取得的结果，来判别该嵌入装置的种类、安装的选项部件的种类、该嵌入装置以及选项部件的各版本或该嵌入装置以及选项部件的各状态，显示与该判别结果对应的文书的内容。

26. 一种嵌入装置的控制方法，其特征在于，包括：

加载至少包含由标记语言记述的文书数据的内容数据，根据该加载的内容数据来进行显示的浏览器步骤；

控制该嵌入装置的信息和动作的控制步骤；

提供用于所述浏览器步骤访问由所述控制步骤控制的该嵌入装置的信息和动作的接口的接口步骤；

所述浏览器步骤处理嵌入到上述文书数据中的脚本或者由嵌入到上述文书数据中的参照信息所参照的脚本，根据该处理通过所述接口步骤访问由所述控制步骤控制的该嵌入装置的信息和动作，根据通过该访问而取得的结果，来判别该嵌入装置中是否存在内部程序，如果存在内部程序，就显示包含用于指示该内部程序的起动的用户输入要素的文书。

27. 一种嵌入装置的控制方法，其特征在于，包括：

控制该嵌入装置的信息和动作的控制步骤；

处理脚本的脚本处理步骤；

提供用于所述脚本处理步骤根据所述脚本的处理访问由所述控制步骤控制的该嵌入装置的信息和动作的接口的接口步骤；

所述接口步骤构筑所述脚本处理步骤用于访问由所述控制步骤控制的该嵌入装置的信息和动作的装置对象模型，通过把构成该嵌入装置的各种资源分别与所述脚本处理步骤作为处理对象的各对象关联，把所述各资源的属性与该资源所对应的对象的属性关联，把对于所述各资源的操作与该资源所对应的对象的操作关联，把所述各资源间的关系与该资源间所对应的对象间的关系关联，来构筑所述装置对象模型。

嵌入装置及其控制方法

技术领域

本发明涉及具有 Web 浏览器的嵌入 (embedded) 装置、它的控制方法、用于实现该控制方法的程序和存储该程序的存储媒体。

背景技术

众所周知, 通过由 HTTP (Hypertext Transfer Protocol) 供给由 HTML (Hypertext Markup Language) 表现的文本数据的 Web 服务器、从 Web 服务器通过 HTTP 取得 HTTP 数据并且显示的客户机即 Web 浏览器, 因特网取得了发展。

Web 浏览器的最基本的功能作为对从客户机通过 HTTP 发送的文件的取得要求的响应, 发回要求的文件的文件传输服务。在此, 基本上只读出该服务器中存储的静态文件, 发回, 但是除了该基本功能, 有时还按照来自客户机的 HTTP 要求 (GET 要求和 POST 要求), 动态地生成数据并发回。

Web 服务器动态生成 HTTP 响应的处理称作服务器一侧处理。有时委托给按照 CGI (Common Gateway Interface) 规格、用于 Java (注册商标) 的 Servlet 规格安装的外部软件, 实现服务器一侧处理。按照这些接口规格从 Web 服务器委托处理的外部软件作为服务器的执行形式程序、由 Java 等虚拟机用的命令语言记述的程序或由解释程序执行的文本形式的脚本安装。作为由解释程序 (脚本执行环境) 执行的文本形式的脚本的具体例, 使用 shell script、JavaScript、PHP (Personal HomePage 或 PHP: Hypertext Preprocessor) 等脚本。把为了服务器一侧处理提供与 Web 服务器联合的脚本称作服务器一侧脚本。

通过 HTML 的 FORM 要素, HTML 文书能表现用于要求来自用户的输入的格式。基于 FORM 要素的格式在 Web 浏览器上显示, 用户进行对格式的输入, 如果用户指示提交, 则对格式输入的数据作为 HTTP 要求 (GET 要求或 POST 要求) 发送给 Web 服务器一侧。因此, 通过包含能接受用于对服务器一侧处理的指示的格式地构成 Web 服务器发送给客户机的 HTML 文书, 能提供用户通过 Web 服务器与服务器一侧处理对话的一种用户界面 (也能把用户在格式内输入

指定的客户一侧的文件上传给服务器一侧地构成格式)。服务器一侧处理不仅动态生成 HTTP 响应,作为副作用,能执行用于对用户提供的任意的服务的处理。

用户能通过 Web 服务器利用存在于服务器一侧的业务逻辑和后端服务等。常常使用把 HTML 作为一种用户界面记述语言使用,只把使处理的大部分由服务器一侧执行的服务的用户界面处理由客户机一侧的 Web 浏览器担当的分散系统的模型(把这样的模型中的轻量的客户机称作轻客户机 (Thin Client))。

而也知道在 Web 浏览器一侧进行计算处理的客户机一侧处理。在客户机一侧处理中,按照来自 Web 浏览器的要求,把用于进行任意处理的软件从 Web 服务器发回客户机一侧,在嵌入 Web 浏览器中的处理系统上执行该软件。此时,动态的文书的生成或与用户的对话由在 Web 浏览器中工作的客户机一侧处理实现,没必要一定与服务器一侧通信。由 Java 虚拟机用的命令语言记述,按照 Java Applet 的接口规格安装的对象形式的程序在作为插件程序嵌入 Web 浏览器中的 Java 虚拟机上工作。另外,作为基于文本的脚本语言,此前在很多 Web 浏览器中嵌入 JavaScript 解释程序。客户机一侧 JavaScript 组合由 Web 浏览器定义的 DOM (Document Object Model) 和 JavaScript 解释程序的脚本功能。与 Java Applet 等的方法相比,基于 JavaScript 等的脚本语言的方法是基于文本,所以与 HTML 等标记语言的亲和性高,具有 HTML 文书内容的提供者能简便提供“可执行的内容”的长处。另外,客户机一侧 JavaScript 的解释程序与 Web 浏览器用于在内部处理文书的数据构造 DOM 联合构成,所以能操作 Web 浏览器中显示的文书,简单地对用户提供的动态的信息和用户界面。

JavaScript 语言的核心部分在 ECMA (European Computer Manufacturers Association) 中标准化为 ECMAScript。客户机一侧服务和 JavaScript 的细节例如参照“JavaScript The Definitive Guide, Third Edition”, David Flanagan, O'Reilly, 1998 等。

DOM 是用于把用 HTML 等标记语言记述的文书的构造作为对象处理的对象模型。Netscape Navigator (注册商标) 和 Internet Explorer(注册商标)等各 Web 浏览器为了 JavaScript 等客户机一侧处理,从以往就在各自的安装中提供固有的 DOM。DOM 规格由 W3C (World Wide Web Consortium) 进行标准化,能利用依据标准规格的接口。

另外,在通用的桌面计算机用的 Web 浏览器中,通过取入插件软件、动态链接库或基于本地机械代码的对象,动态地与 Web 浏览器

的执行形式结合，扩展 Web 浏览器自身的功能。

例如在特开平 11-187061 号公报中描述基于服务器一侧处理 (CGI) 的装置控制。

另外，在特开平 11-134125 号公报、特开平 11-212751 号公报、特开平 11-327834 号公报和特开 2000-194531 号公报中描述内置在数字复合机中的 Web 共享打印功能。Web 共享打印功能是具有与 Web 浏览器具有的机构同样的基于 HTTP 的数据取得机构和 HTML 等的表现机构，打印输出从用户指定的 URL (Uniform Resource Locator) 取得的内容的功能。

一般知道在各种嵌入装置 (嵌入 (incorporate) 了操作系统或软件的装置) 中，在固件中设置 Java 虚拟机，从服务器取得由虚拟机的对象形式 (字节代码) 记述的程序，能动态加载、执行的系统。

但是，在嵌入装置的固件中嵌入的 Web 浏览器中要求与通用的桌上计算机用的 Web 浏览器不同的应用。这是因为一般嵌入装置的计算资源 (CPU、存储器、外部存储、显示器的尺寸和析像度、键盘和定点设备) 常常比通用的桌上计算机在功能和性能上差。另外，通用的桌上计算机是为了完成各种工作，放在桌子上，用户坐在椅子上操作的计算机，如果是数字复制机等复合机，就是放在办公室的一角，多个成员公用，在特定的情况和利用形态中利用的。使用嵌入装置中内置的 Web 浏览器，不能进行一般的因特网的网上冲浪和浏览，不能说是最佳的用途。

因此，在嵌入装置的固件中嵌入的 Web 浏览器的应用中，要求与嵌入装置本来固有的功能联动，浏览最佳的内容。

作为嵌入浏览器的应用事例，有耗材销售、手册的参照、装置诊断等服务的利用。例如嵌入复合机中的 Web 浏览器在显示装置的销售商提供的入口页时，如果该装置的粉末残余量小，就动态显示推荐粉末的再现购买的引导等与顾客环境中固有的装置利用密切相关的极其细致的服务的提供是有效的。另外，从嵌入装置的 Web 浏览器显示公开该装置的最新手册、事例集、专门知识的页时，有选择地显示关于该装置或装置中现在安装的选项群的信息等极其细致的服务的提供是有效的。在进行装置的维护和修理时，嵌入装置中的固件常常必须参照或变更作为内部数据管理的多种多样的参数，以适当的顺序和组合调用固件中设置的多个诊断代码和安装代码。而且，有时根据选项部件的安装状况和固件的版本，恰当的处置不同。伴随着专门知识的积累，有时最佳的处置变更。因此，最终用户和现场的客户工程师把握全部，进行最佳的处置是困难的或需要很大的成本。关于这样的装置

的管理作业，在顾客环境中如果从装置的嵌入浏览器访问服务公司提供的装置管理信息的网页，就能观察动态反映该装置的现状的最佳作业手册，如果按照装置的种类和状况，就能显示服务公司内部按照最近的专业知识周到地准备的用于进行装置内部的参数的参照和变更或诊断代码和安装代码的组的执行指示等的用户界面页，就极其方便。

从这些例子可知，希望通过服务器一侧的对应以能灵活扩张的形式提供最适合来自嵌入浏览器的访问的与装置的功能或状态联动的动态内容。但是，从隐私的观点出发，有时不希望对服务器一侧（内容提供者一侧）通知装置的信息。此时，使用 CGI 程序的服务器一侧处理不适合，所以有必要通过客户机一侧处理实现。

另外，必须同时考虑从外部经由网络取得的内容能充分与装置的功能或状态联合和不威胁嵌入装置的安全性。

嵌入装置中内置的浏览器几乎都是作为实时 OS 上的固件安装，所以与通用桌上计算机用的浏览器不同，难以取入插件软件、动态链接库或基于本地机械代码的对象，动态地与浏览器的执行形式结合，实现浏览器自身的功能扩张。因此，浏览器的功能在系统的出厂时预先固定，并且要求通过在服务器一侧以能灵活扩张的形式实现与装置的功能或状态联合的动态内容的提供。

但是，在所述特开平 11-187061 号公报中，记载了基于服务器一侧处理（CGI）的装置控制，但是不触及基于所述嵌入浏览器上的客户机一侧处理的装置控制。

另外，基于 Java 虚拟机的方法如上所述，在与 HTML 等基于文本的标记语言的亲和性、从程序对嵌入该程序（或关联）的原来的文书的 DOM 的访问性上，比基于 JavaScript 等脚本的方法差，所以对于用于提供利用装置嵌入浏览器的分散系统的内容提供者不够简便。

发明内容

鉴于上述问题的存在，本发明的目的在于：提供与来自嵌入浏览器的访问对应，能把与嵌入装置固有的功能或状态联动适应的动态内容作为服务器一侧能简单灵活地扩张的服务供给，保护客户机一侧信息的隐私和安全性的 Web 浏览器内置嵌入装置、它的控制方法、用于实现该控制方法的程序和存储该程序的存储媒体。

本发明提供一种嵌入装置，其特征在于，包括：加载至少包含由标记语言记述的文书数据即其中嵌入脚本或参照脚本的信息的文书数据的内容数据，并根据该加载的内容数据来进行显示的浏览器部件；控制该嵌入装置的信息和动作的控制部件；解释所述加载的内容数据

中包含的文书数据，并处理嵌入该文书数据中的脚本或由参照信息参照的脚本的脚本处理部件，即嵌入所述浏览器部件中的脚本处理部件；提供用于所述脚本处理部件根据所述脚本的处理来访问由所述控制部件控制的该嵌入装置的信息和动作的接口的第一接口部件；和提供用于所述脚本处理部件根据所述脚本的处理来访问在所述文书数据的内部数据构造的接口的第二接口部件，所述脚本处理部件把根据通过所述第一接口部件访问由所述控制部件控制的该嵌入装置的信息和动作而取得的结果所生成的一部分文书数据插入所述文书数据中的与通过所述第二接口部件访问所述文书数据的内部数据构造而取得的结果对应的位置。

本发明提供一种嵌入装置，其特征在于，包括：加载至少包含由标记语言记述的文书数据的内容数据，并根据该加载的内容数据来进行显示的浏览器部件；控制该嵌入装置的信息和动作的控制部件；提供所述浏览器部件用于访问由所述控制部件控制的该嵌入装置的信息和动作的接口的接口部件；所述浏览器部件处理嵌入到上述文书数据中的脚本或者由嵌入到上述文书数据中的参照信息所参照的脚本，根据该处理通过所述接口部件访问由所述控制部件控制的该嵌入装置的信息和动作，根据通过该访问而取得的结果，识别正在使用该嵌入装置的用户，显示与该用户对应的文书的内容。

本发明提供一种嵌入装置，其特征在于，包括：加载至少包含由标记语言记述的文书数据的内容数据，并根据该加载的内容数据来进行显示的浏览器部件；控制该嵌入装置的信息和动作的控制部件；提供所述浏览器部件用于访问由所述控制部件控制的该嵌入装置的信息和动作的接口的接口部件；所述浏览器部件处理嵌入到上述文书数据中的脚本或者由嵌入到上述文书数据中的参照信息所参照的脚本，根据该处理通过所述接口部件访问由所述控制部件控制的该嵌入装置的信息和动作，根据通过该访问而取得的结果，来判别该嵌入装置的固有信息或该嵌入装置的状态，显示与该判别结果对应的文书的内容。

本发明提供一种嵌入装置，其特征在于，包括：加载至少包含由标记语言记述的文书数据的内容数据，并根据该加载的内容数据来进行显示的浏览器部件；控制该嵌入装置的信息和动作的控制部件；提供所述浏览器部件用于访问由所述控制部件控制的该嵌入装置的信息和动作的接口的接口部件；所述浏览器部件处理嵌入到上述文书数据中的脚本或者由嵌入到上述文书数据中的参照信息所参照的脚本，根据该处理通过所述接口部件访问由所述控制部件控制的该嵌入装置的信息和动作，根据通过该访问而取得的结果，来判别该嵌入装置中是

否存在内部程序，如果存在内部程序，就显示包含用于指示该内部程序的起动的用户输入要素的文书。

本发明提供一种嵌入装置，其特征在于，包括：控制该嵌入装置的信息和动作的控制部件；处理脚本的脚本处理部件；提供所述脚本处理部件根据所述脚本的处理用于访问由所述控制部件控制的该嵌入装置的信息和动作的接口的接口部件；所述接口部件构筑用于所述脚本处理部件访问由所述控制部件控制的该嵌入装置的信息和动作的装置对象模型，通过把构成该嵌入装置的各种资源分别与所述脚本处理部件作为处理对象的各对象关联，把所述各资源的属性与该资源所对应的对象的属性关联，把对于所述各资源的操作与该资源所对应的对象的操作关联，把所述各资源间的关系与该资源间所对应的对象间的关系关联，来构筑所述装置对象模型。

本发明提供一种嵌入装置的控制方法，其特征在于，包括：加载至少包含由标记语言记述的文书数据即其中嵌入脚本或参照脚本的信息的文书数据的内容数据，根据该加载的内容数据来进行显示的浏览器步骤；控制该嵌入装置的信息和动作的控制步骤；解释所述加载的内容数据中包含的文书数据，并处理嵌入该文书数据中的脚本或由参照信息参照的脚本的脚本处理步骤，即嵌入所述浏览器步骤中的脚本处理步骤；提供用于所述脚本处理步骤根据所述脚本的处理来访问由所述控制步骤控制的该嵌入装置的信息和动作的接口的第一接口步骤；和提供用于在所述脚本处理步骤中的根据所述脚本的处理来访问所述文书数据的内部数据构造的接口的第二接口步骤，所述脚本处理步骤把根据通过所述第一接口步骤访问由所述控制步骤控制的该嵌入装置的信息和动作而取得的结果所生成的一部分文书数据插入所述文书数据中的与通过所述第二接口步骤访问所述文书数据的内部数据构造而取得的结果对应的位置。

本发明提供一种嵌入装置的控制方法，其特征在于，包括：加载至少包含由标记语言记述的文书数据的内容数据，根据该加载的内容数据来进行显示的浏览器步骤；控制该嵌入装置的信息和动作的控制步骤；提供所述浏览器步骤用于访问由所述控制步骤控制的该嵌入装置的信息和动作的接口的接口步骤；所述浏览器步骤处理嵌入到上述文书数据中的脚本或者由嵌入到上述文书数据中的参照信息所参照的脚本，根据该处理通过所述接口步骤访问由所述控制步骤控制的该嵌入装置的信息和动作，根据通过该访问而取得的结果，识别正在使用该嵌入装置的用户，显示与该用户对应的文书的内容。

本发明提供一种嵌入装置的控制方法，其特征在于，包括：加载

至少包含由标记语言记述的文书数据的内容数据,根据该加载的内容数据来进行显示的浏览器步骤;控制该嵌入装置的信息和动作的控制步骤;提供所述浏览器步骤用于访问由所述控制步骤控制的该嵌入装置的信息和动作的接口的接口步骤;所述浏览器步骤处理嵌入到上述文书数据中的脚本或者由嵌入到上述文书数据中的参照信息所参照的脚本,根据该处理通过所述接口步骤访问由所述控制步骤控制的该嵌入装置的信息和动作,根据通过该访问而取得的结果,来判别该嵌入装置的种类、安装的选项部件的种类、该嵌入装置以及选项部件的各版本或该嵌入装置以及选项部件的各状态,显示与该判别结果对应的文书的内容。

本发明提供一种嵌入装置的控制方法,其特征在于,包括:加载至少包含由标记语言记述的文书数据的内容数据,根据该加载的内容数据来进行显示的浏览器步骤;控制该嵌入装置的信息和动作的控制步骤;提供用于所述浏览器步骤访问由所述控制步骤控制的该嵌入装置的信息和动作的接口的接口步骤;所述浏览器步骤处理嵌入到上述文书数据中的脚本或者由嵌入到上述文书数据中的参照信息所参照的脚本,根据该处理通过所述接口步骤访问由所述控制步骤控制的该嵌入装置的信息和动作,根据通过该访问而取得的结果,来判别该嵌入装置中是否存在内部程序,如果存在内部程序,就显示包含用于指示该内部程序的起动的用户输入要素的文书。

本发明提供一种嵌入装置的控制方法,其特征在于,包括:控制该嵌入装置的信息和动作的控制步骤;处理脚本的脚本处理步骤;提供用于所述脚本处理步骤根据所述脚本的处理访问由所述控制步骤控制的该嵌入装置的信息和动作的接口的接口步骤;所述接口步骤构筑所述脚本处理步骤用于访问由所述控制步骤控制的该嵌入装置的信息和动作的装置对象模型,通过把构成该嵌入装置的各种资源分别与所述脚本处理步骤作为处理对象的各对象关联,把所述各资源的属性与该资源所对应的对象的属性关联,把对于所述各资源的操作与该资源所对应的对象的操作关联,把所述各资源间的关系与该资源间所对应的对象间的关系关联,来构筑所述装置对象模型。

为了实现所述目的,根据本发明的第一方面,提供一种嵌入装置,包括:加载至少包含由标记语言记述的文书数据即其中嵌入脚本或参照脚本的信息的文书数据的内容数据,并根据该加载的内容数据来进行显示的浏览器部件;控制该嵌入装置的信息和动作的控制部件;解释所述加载的内容数据中包含的文书数据,并处理嵌入该文书数据中的脚本或由参照信息参照的脚本的脚本处理部件,即嵌入所述浏览器

部件中的脚本处理部件；提供用于所述脚本处理部件根据所述脚本的处理来访问由所述控制部件控制的该嵌入装置的信息和动作的接口的第一接口部件。

根据本发明的第一方面的嵌入装置的结构，脚本处理部件解释加载的 Web 内容中包含的文书数据，处理该文书数据中嵌入的脚本或通过参照信息参照的脚本，根据该脚本的处理，第一接口部件能访问在浏览器部件内管理的文书数据的内部数据构造，第二接口部件使所述脚本处理部件根据所述脚本的处理能访问由控制部件控制的该嵌入装置的信息和动作。通过第一和第二接口生成与嵌入装置固有的功能或状态联动适应的动态内容，根据该内容进行显示，所以只在服务器一侧准备单一的内容，就能有选择地按照嵌入装置的结构或状态提示信息。另外，动态内容的生成主要由客户机一侧处理进行，所以能保护客户机一侧信息的隐私和安全性。

更好是还具有提供用于所述脚本处理部件根据所述脚本的处理来访问在所述浏览器部件内管理的所述文书数据的内部数据构造的接口的第二接口部件，所述脚本处理部件把根据通过所述第一接口部件访问由所述控制部件控制的该嵌入装置的信息和动作而取得的结果而生成的一部分文书数据插入所述文书数据中的与通过所述第二接口部件访问所述文书数据的内部数据构造而取得的结果对应的位置。

更好是所述第二接口部件构筑用于所述脚本处理部件访问所述文书数据的内部数据构造的文档对象模型，把所述文书数据的各要素分别与所述脚本处理部件作为处理对象的各对象关联，把所述文书数据的各要素的属性与该要素所对应的对象的属性关联，把对于所述文书数据的各要素的操作与对于该要素所对应的对象的操作关联，把所述文书数据的各要素之间的关系与对应于该要素间的对象间的关系关联，把对于所述文书数据或其要素生成的事件与对于该文书数据或要素所对应的对象生成的事件关联，来构筑所述文档对象模型。

更好是所述脚本处理部件与对于所述文书数据或其要素生成的事件呼应，通过所述第一接口部件访问由所述控制部件控制的该嵌入装置的信息和动作。

更好是所述第一接口部件构筑用于所述脚本处理部件访问由所述控制部件控制的该嵌入装置的信息和动作的装置对象模型，把构成该嵌入装置的各种资源分别与所述脚本处理部件作为处理对象的各对象关联，把所述各资源的属性与该资源所对应的对象的属性关联，把对于所述各资源的操作与该资源所对应的对象的操作关联，把所述各资源间的关系与该资源间所对应的对象间的关系关联，来构筑装置对

象模型。

更好是在所述装置对象模型中，继承抽象-具体关系，具体类具有抽象类的属性和操作中至少一方的至少一部分，属于各类的对象具有该类的属性和操作中至少一方的至少一部分。

更好是在所述装置对象模型中，表现全体-部分的聚合关系，全体对象具有用于探索与该全体对象关联的0个以上的部分对象的操作或属性。

更好是把对于该嵌入装置或其资源生成的事件和例外分别与对于该嵌入装置或其资源所对应的对象生成的事件和例外关联，来构筑所述装置对象模型。

更好是，当由用户指示把所述脚本处理部件通过所述第一接口部件访问由所述控制部件控制的该嵌入装置的信息和动作而取得的结果向该嵌入装置的外部输出时，所述浏览器部件禁止与该指示对应的外部输出。

更好是所述脚本处理部件当由用户指示把所述脚本处理部件通过所述第一接口部件访问由所述控制部件控制的该嵌入装置的信息和动作而取得的结果向该嵌入装置的外部输出时，所述浏览器部件预先或在指示时对用户确认与该指示对应的外部输出的可否。

更好是按照构成所述装置对象模型各对象，保持访问权限，所述脚本处理部件尝试对没有访问权限的对象、对象的属性或对象的操作的访问时，所述第一接口部件禁止该访问。

更好是具有：设定对于构成所述装置对象模型各对象的访问权限的设定部件。

更好是构成所述装置对象模型各对象继承该对象所属的类中保持的访问权限。

更好是作为构成装置对象模型各对象的访问权限，设定与该对象存在聚合关系，并且包含的对象中保持的访问权限。

为了实现所述目的，根据本发明的第二方面，提供一种嵌入装置，包括：加载至少包含由标记语言记述的文书数据的内容数据，并根据该加载的内容数据来进行显示的浏览器部件；控制该嵌入装置的信息和动作的控制部件；提供所述浏览器部件用于访问由所述控制部件控制的该嵌入装置的信息和动作的接口的接口部件；所述浏览器部件根据通过所述接口部件访问由所述控制部件控制的该嵌入装置的信息和动作而取得的结果，识别正在使用该嵌入装置的用户，显示与该用户对应的文书的内容。

为了实现所述目的，根据本发明的第三方面，提供一种嵌入装置，

包括：加载至少包含由标记语言记述的文书数据的内容数据，并根据该加载的内容数据来进行显示的浏览器部件；控制该嵌入装置的信息和动作的控制部件；提供所述浏览器部件用于访问由所述控制部件控制的该嵌入装置的信息和动作的接口的接口部件；所述浏览器部件根据通过所述接口部件访问由所述控制部件控制的该嵌入装置的信息和动作而取得的结果，来判别该嵌入装置的固有信息或该嵌入装置的状态，显示与该判别结果对应的文书的内容。

根据本发明的第三方法的嵌入装置的结构，根据通过接口部件访问由控制部件控制的该嵌入装置的信息和动作而取得的结果，来判别该嵌入装置的种类、安装的选项部件的种类、该嵌入装置以及选项部件的各版本、或该嵌入装置以及选项部件的各状态，显示与该判别结果对应的文书的内容，所以能在为了提供由浏览器浏览的最适合于该装置状态的电子手册时利用。

更好是所述固有信息包含该嵌入装置的种类、安装的选项的种类、该嵌入装置以及选项部件的各版本中至少任意一个。

更好是所述文书包含关于该嵌入装置以及该嵌入装置中安装的选项部件的至少一个的手册、对用户的提议、参照所述手册或所述提议的信息的至少一个。

为了实现所述目的，根据本发明的第四方面，提供一种嵌入装置，包括：加载至少包含由标记语言记述的文书数据的内容数据，并根据该加载的内容数据来进行显示的浏览器部件；控制该嵌入装置的信息和动作的控制部件；提供所述浏览器部件用于访问由所述控制部件控制的该嵌入装置的信息和动作的接口的接口部件；所述浏览器部件根据通过所述接口部件访问由所述控制部件控制的该嵌入装置的信息和动作而取得的结果，来判别该嵌入装置中是否存在内部程序，如果存在内部程序，就显示包含用于指示该内部程序的起动的用户输入要素的文书。

为了实现所述目的，根据本发明的第五方面，提供一种嵌入装置，包括：控制该嵌入装置的信息和动作的控制部件；处理脚本的脚本处理部件；提供所述脚本处理部件根据所述脚本的处理用于访问由所述控制部件控制的该嵌入装置的信息和动作的接口的接口部件；所述接口部件构筑用于所述脚本处理部件访问由所述控制部件控制的该嵌入装置的信息和动作的装置对象模型，通过把构成该嵌入装置的各种资源分别与所述脚本处理部件作为处理对象的各对象关联，把所述各资源的属性与该资源所对应的对象的属性关联，把对于所述各资源的操作与该资源所对应的对象的操作关联，把所述各资源间的关系与该资

源间所对应的对象间的关系关联，来构筑所述装置对象模型。

更好是在所述装置对象模型中，继承抽象-具体关系，具体类具有抽象类的属性和操作中至少一方的至少一部分，属于各类的对象具有该类的属性和操作中至少一方的至少一部分。

更好是在所述装置对象模型中，表现全体-部分的聚合关系，全体对象具有用于探索与该全体对象关联的0个以上的部分对象的操作或属性。

更好是把对于该嵌入装置或其资源生成的事件和例外分别与对于该嵌入装置或其资源所对应的对象生成的事件和例外关联，来构筑所述装置对象模型。

为了实现所述目的，根据本发明的第六方面，提供一种嵌入装置的控制方法包括：加载至少包含由标记语言记述的文书数据即其中嵌入脚本或参照脚本的信息的文书数据的内容数据，根据该加载的内容数据来进行显示的浏览器步骤；控制该嵌入装置的信息和动作的控制步骤；解释所述加载的内容数据中包含的文书数据，并处理嵌入该文书数据中的脚本或由参照信息参照的脚本的脚本处理步骤，即嵌入所述浏览器步骤中的脚本处理步骤；提供所述脚本处理步骤根据所述脚本的处理来访问由所述控制步骤控制的该嵌入装置的信息和动作的接口的接口步骤。

为了实现所述目的，根据本发明的第七方面，提供一种嵌入装置的控制方法包括：加载至少包含由标记语言记述的文书数据的内容数据，根据该加载的内容数据来进行显示的浏览器步骤；控制该嵌入装置的信息和动作的控制步骤；提供所述浏览器步骤用于访问由所述控制步骤控制的该嵌入装置的信息和动作的接口的接口步骤；所述浏览器步骤根据通过所述接口步骤访问由所述控制步骤控制的该嵌入装置的信息和动作而取得的结果，识别正在使用该嵌入装置的用户，显示与该用户对应的文书的内容。

为了实现所述目的，根据本发明的第八方面，提供一种嵌入装置的控制方法，包括：加载至少包含由标记语言记述的文书数据的内容数据，根据该加载的内容数据来进行显示的浏览器步骤；控制该嵌入装置的信息和动作的控制步骤；提供所述浏览器步骤用于访问由所述控制步骤控制的该嵌入装置的信息和动作的接口的接口步骤；所述浏览器步骤根据通过所述接口步骤访问由所述控制步骤控制的该嵌入装置的信息和动作而取得的结果，来判别该嵌入装置的种类、安装的选项部件的种类、该嵌入装置以及选项部件的各版本或该嵌入装置以及选项部件的各状态，显示与该判别结果对应的文书的内容。

为了实现所述目的,根据本发明的第九方面,提供一种嵌入装置的控制方法,包括:加载至少包含由标记语言记述的文书数据的内容数据,根据该加载的内容数据来进行显示的浏览器步骤;控制该嵌入装置的信息和动作的控制步骤;提供所述浏览器步骤用于访问由所述控制步骤控制的该嵌入装置的信息和动作的接口的接口步骤;所述浏览器步骤根据通过所述接口步骤访问由所述控制步骤控制的该嵌入装置的信息和动作而取得的结果,来判别该嵌入装置中是否存在内部程序,如果存在内部程序,就显示包含用于指示该内部程序的起动的用户输入要素的文书。

为了实现所述目的,根据本发明的第十方面,提供一种嵌入装置的控制方法,包括:控制该嵌入装置的信息和动作的控制步骤;处理脚本的脚本处理步骤;提供所述脚本处理步骤根据所述脚本的处理用于访问由所述控制步骤控制的该嵌入装置的信息和动作的接口的接口步骤;所述接口步骤构筑所述脚本处理步骤用于访问由所述控制步骤控制的该嵌入装置的信息和动作的装置对象模型,通过把构成该嵌入装置的各种资源分别与所述脚本处理步骤作为处理对象的各对象关联,把所述各资源的属性与该资源所对应的对象的属性关联,把对于所述各资源的操作与该资源所对应的对象的操作关联,把所述各资源间的关系与该资源间所对应的对象间的关系关联,来构筑所述装置对象模型。

从以下的参照附图的详细说明中,本发明的以上和其它目的、特征和优势将变得明显。

附图说明

图1是表示包含应用本发明实施例1的嵌入装置的图象处理装置的图象处理系统的全体结构的框图。

图2是表示图1的图象处理装置的软件结构的框图。

图3是表示图1的图象处理装置的详细硬件结构的框图。

图4是表示图1的图象处理装置的外观的图。

图5是表示图4中的操作部的详细外观的立体图。

图6是表示图5的操作部的详细功能结构的框图。

图7是表示用于说明嵌入图1的图象处理装置中的应用程序的动作的网络结构一例的框图。

图8是表示图2中的Web浏览器模块的软件结构的框图。

图9是表示图5中的LCD显示部中显示的Web浏览器的画面结构的框图。

图 10 是表示基于 HTTP 的要求和响应的处理流程的一例的序列图。

图 11 是表示服务器一侧处理和客户机一侧处理的流程一例的图。

图 12 是表示管理利用图 1 的图象处理装置的用户的用户账户的数据构造一例的图。

图 13 是表示管理利用图 1 的图象处理装置的用户的用户组的数据构造一例的图。

图 14 是表示把构成图 1 的图象处理装置的各种资源抽象化表现的资源类的数据构造一例的模式图。

图 15A、15B、15C 是表示代表构成图 1 的图象处理装置的各种资源间的继承关系的数据构造一例的类图。

图 16 是表示代表图 1 的图象处理装置处理的各种资源间的聚合关系的数据构造一例的类图。

图 17 是表示构成装置对象模型的实例的对象的一部分的图。

图 18 是表示嵌入 Web 浏览器中嵌入的 JavaScript 翻译程序作为脚本的动作环境提供的客户机一侧对象的全体-部分阶层构造的一例的图。

图 19 是表示通过 Web 浏览器从服务器取得的文书的一例的图。

图 20 是表示接着图 19 的文书的文书的图。

图 21 是动态适应的文书的 Web 浏览器上的显示例。

图 22 是动态适应的文书的 Web 浏览器上的其他显示例。

图 23 是按照由脚本处理动态生成的 URL 起动的电子邮件发送功能的画面例。

图 24 是表示通过应用本发明实施例 2 的嵌入装置的图象处理装置的 Web 浏览器从服务器取得的文书的一例的图。

图 25 是表示接着图 24 的文书的文书的图。

图 26 是表示接着图 25 的文书的文书的图。

图 27 是动态适应的手册文书的 Web 浏览器上的显示例。

图 28 是表示通过应用本发明实施例 3 的嵌入装置的图象处理装置的 Web 浏览器从服务器取得的文书的一例的图。

图 29 是表示接着图 28 的文书的文书的图。

图 30 是动态适应的装置管理文书的 Web 浏览器上的显示例。

图 31 是取得文书时的显示结束后新动态地变更的文书的 Web 浏览器上的显示例。

图 32 是取得文书时的显示结束后新动态地变更的文书的 Web 浏览器上的其他显示例。

图 33 是表示通过应用本发明实施例 4 的嵌入装置的图象处理装置的 Web 浏览器从服务器取得的文书的一例的图。

图 34 是表示接着图 33 的文书的文书的图。

图 35 是动态的装置报告文书的 Web 浏览器上的显示例。

图 36 是动态适应的文书的 Web 浏览器上的显示例。

图 37 是表示 HDD 的逻辑使用方法的图。

图 38 是作为资源的一例，表示参照和设定文书资源实例的访问全的用户界面的一例的图。

图 39 是作为资源的其他一例，表示参照和设定彩色复制功能资源实例的访问权的用户界面的一例的图。

图 40 是表示显示资源间的继承关系，参照和设定彩色复制功能资源类的访问权的用户界面的一例的图。

图 41 是表示显示资源间的继承关系，参照和设定复制功能资源类的访问权的用户界面的一例的图。

图 42 是表示显示资源间的继承关系，参照和设定黑白复制功能资源类的访问权的用户界面的一例的图。

图 43 是表示显示资源间的聚合关系，参照和设定打印部资源类的访问权的用户界面的一例的图。

图 44 是表示显示资源间的聚合关系，参照和设定排纸资源类的访问权的用户界面的一例的图。

图 45 是表示违反访问权错误显示一例的图。

具体实施方式

下面，参照附图来详细说明本发明实施例。

图 1 是表示包含应用本发明实施例 1 的嵌入装置的图象处理装置的图象处理系统的全体结构的框图。

本图象处理系统由应用服务提供者 (ASP) 站点 153、广域网 152 和用户站点 151 构成。

作为广域网 152 的一例，是因特网。另外，可以是因特网上的虚拟个人网络、专用的个人网络。

ASP 站点 153 通过广域网 152 对用户站点 151 提供给定的服务。作为 ASP 站点 153 提供的服务，有信息的提供、生成、检索、存储、认证、发布、打印、出版、管理、翻译、委托等。另外，也有官厅和地方公共机关手续或各种电子商务。

ASP 站点 153 包含局域网 (LAN) 154 和服务服务器 155。

LAN154 是 ASP 站点 153 内部的网络，连接 ASP 站点 153 内的

网络仪器。另外，LAN154 通过未图示的路由器与广域网 152 连接。

在服务器 155 上，用于实现 ASP 提供的服务的软件进程群工作。在构成该软件进程群的软件模块中包含以下。即响应来自客户机的基于 HTTP 的要求，传送 HTML 等内容的 HTML 服务器；按照 HTTP 要求由 HTTP 服务器执行，进行给定处理动态变化的 HTTP 响应的 CGI 程序或 Servlet 等形态中安装的 Web 应用程序群；CGI 程序或 Servlet 为了执行给定的处理而使用的电子商务程序等的交易逻辑群和后端的数据库管理系统。

用户站点 151 由主机 101、图象处理装置 110、120 以及 130 等多个网络仪器、连接网络仪器群的 LAN100 构成。用户站点 151 的 LAN100 通过未图示的路由器连接在广域网 152 上。路由器页具有所谓的防火墙 (firewall) 的功能。即为了从来自外部网络的攻击保护用户站点 151，进行信息报过滤。另外，为了地址管理上的理由等，有时也进行网络地址的变换或网络端口的变换。为了路由器的这些功能，对用户站点 151 和外部网络之间的通信设置限制。即常常只由基于几个特定的协议的通信成为可能。例如从内部向外部确立的 HTTP 连接是一般允许的通信的例子，它成为基于这样一般的 Web 页技术的应用服务的提供是有效的理由之一。

须指出的是，图象处理装置 110、120 和 130 都相当于本实施例的嵌入装置，但是各图象处理装置 110、120 和 130 分别同样构成，所以在说明时，只把图象处理装置 110 作为本实施例的嵌入装置。

图象处理装置 110 是进行图象的输入输出和收发以及各种图象处理的复合机 (MFP: Multi Function Peripheral)，由图象输入设备及读出部 113、图象输出设备及打印部 114、控制部件 (Controller Unit) 111 和用户接口即操作部 112 构成。

读出部 113、打印部 114 和操作部 112 分别连接在控制部件 111 上，由来自控制部件 111 的命令控制。控制部件 111 连接在 LAN100 等的网络传送部件上。

另外，在 LAN100 上连接具有与图象处理装置 110 同样的仪器结构的其他图象处理装置 120 和 130。图象处理装置 120 由读出部 123、打印部 124 和操作部 122 构成，它们连接在控制部件 121 上，进行控制。另外，图象处理装置 130 由读出部 133、打印部 134 和操作部 132 构成，它们连接在控制部件 131 上，进行控制。

另外，主机 101 连接在 LAN100 等网络传送部件上。主机 101 如后所述，具有 Web 浏览器，根据从各图象处理装置 110、120、130 接收的 HTML 文件，显示各图象处理装置 110、120、130 的状态。另外，通过 HTTP 连接在服务器 155 上，接受服务的提供。

图 2 是表示图象处理装置 110 的软件结构的框图。须指出的是，其他图象处理装置 120 以及 130 的软件结构与图象处理装置 110 同样。

用户界面（以下简称作“UI”）模块 201 是在用户进行对于图象处理装置 110 的各种操作和设定时，进行仪器和用户操作的中介的模块。该模块 201 按照用户的操作，对后面描述的各种模块传送输入信息，进行处理的委托或数据的设定。

地址本（Address-Book）模块 202 是管理数据的发送目标和通信目标的数据库模块。地址本模块 202 管理的数据通过来自 UI 模块 201 的操作，进行数据的追加、删除、取得。另外，地址本模块 202 通过用户操作，对后面描述的各种模块提供数据的发送和通信目标信息。

Web 服务器（Web Server）模块 203 根据来自 Web 客户机（例如主机 101）的要求，以 HTTP 提供由 HTML 等记述的信息。

综合发送部（Universal Send）模块 204 是负责数据的发送的模块。该模块 204 把通过 UI 模块 201 由用户指示的数据向同样指示的通信（输出）目标发布。另外，当由用户使用图象处理装置 110 的扫描功能指示发布数据的生成时，通过后面描述的 API 模块 218 使图象处理装置 110 工作，生成数据。

P550 模块 205 是在综合发送部模块 204 内对输出目标指定应用程序时执行的模块。E-mail 模块 206 是在综合发送部模块 204 内对通信目标指定 E-mail 地址时执行的模块。数据库（DB）模块 207 是在综合发送部模块 204 内对输出目标指定数据库时执行的模块。DP 模块 208 是在综合发送部模块 204 内对输出目标指定与图象处理装置 110 同样的图象处理装置时执行的模块。

远程复制扫描（Remote Copy Scan）模块 209 使用图象处理装置 110 的扫描功能读取图象信息，通过把读取的图象信息向由网络连接的其他图象处理装置输出，使用其它图象处理装置进行由图象处理装置 110 单体实现的复制功能的模块。

遥控打印（Remote Copy Print）模块 210 是使用本图象处理装置 110 的打印机功能输出由网络连接的其他图象处理装置中取得的图象

信息，使用其它图象处理装置进行由图象处理装置 110 单体实现的复制功能的模块。

Web 浏览器 (Web Browser) 模块 211 是读入因特网或企业内部互联网上的各种 Web 站点 (主页) 的信息，进行显示的模块。后面描述 Web 浏览器的详细结构。

HTTP 模块 212 是图象处理装置 110 进行基于 HTTP 的通信时使用的模块，使用后面描述的 TCP/IP (Transmission Control Protocol/Internet Protocol) 通信模块 216 对 Web 服务器模块 203 或 Web 浏览器模块 211 提供通信功能。该模块 212 与以 HTTP 为首的 Web 中使用的各种协议对应，也提供基于安全对应协议的通信功能。

LPR 模块 213 使用 TCP/IP 通信模块 216 对综合发送部模块 204 内的打印模块 205 提供通信功能。

SMTP (Simple Mail Transfer Protocol) 模块 214 使用 TCP/IP 通信模块 216 对综合发送部模块 204 内的 E-mail 模块 206 提供通信功能。

SLM (Salutation Manager) 模块 215 使用 TCP/IP 通信模块 216 对综合发送部模块 204 内的数据库模块 207 和 DP 模块 208、远程复制扫描模块 209、遥控复制打印模块 210 提供通信功能。

TCP/IP 通信模块 216 使用后面描述的网络驱动程序 217 对所述各种模块提供网络通信功能。

网络驱动程序 217 控制物理连接在网络上的部分。

控制 API 模块 218 对综合发送部模块 204 等上游模块提供对于后面描述的作业管理器 (Job Manager) 模块 219 等下游模块的接口。据此，减轻上游和下游的模块间的依存关系，能提高各自的挪用性。

作业管理器模块 219 解释从所述各种模块通过控制 API 模块 218 指示的各种处理，对后面描述的各模块 220、224 以及 226 提供指示。另外，作业管理器模块 219 一元化管理图象处理装置 110 内执行的硬件的处理。

多媒体数字信号编解码管理器 (CODEC Manager) 模块 220 在

作业管理器模块 219 指示的处理中，管理和控制数据的各种压缩和展开。

编码器模块 221 使用给定的编码手法压缩通过由作业管理器模块 219 或后面描述的扫描管理器 (Scan Manager) 模块 224 执行的扫描处理读入的数据。

JPEG 多媒体数字信号编解码器 (JPEG (Joint Photographic Expert Group) CODEC) 模块 222 在由作业管理器模块 219 或扫描管理器模块 224 执行的扫描处理、或由打印管理器 (Print- Manager) 模块 226 执行的打印处理中，进行读入的数据的 JPEG 压缩和打印数据的 JPEG 展开处理。

MMR (Modified Modified READ) Modified (MMR CODEC) 模块 223 在由作业管理器模块 219 或扫描管理器模块 224 执行的扫描处理、或由打印管理器模块 226 执行的打印处理中，进行读入的数据的 MMR 压缩和 MMR 展开处理。

信息嵌入图象多媒体数字信号编解码器 (IEI CODEC) 模块 229 在由作业管理器模块 219 或扫描管理器模块 224 执行的扫描处理、或由打印管理器模块 226 执行的打印处理中，进行读入的图象数据中嵌入的信息的译码、向打印图象数据的信息嵌入。使用条形码或数字水印等编码技术进行向图象数据的信息嵌入。另外，把通过像域分离和 OCR (Optical Character Reader) 技术识别图象数据的图象中的文字，变换为文本数据的文字识别也作为一种译码技术支持。把使用光彩图形处理器的从文本向图象数据的变换、变换的图象数据和原来图象数据的重合 (覆盖) 也作为一种编码技术 (信息嵌入技术) 支持。

扫描管理器 (Scan Manager) 模块 224 管理和控制作业管理器模块 219 指示的扫描管理。

SCSI (Small Computer System Interface) 驱动程序 225 负责扫描管理器模块 224、内部连接在图象处理装置 110 中的读出部 113 之间的通信。

打印管理器 (Print Manager) 模块 226 管理和控制作业管理器

模块 219 指示的打印处理。

引擎接口 (Engine I/F) 模块 227 提供打印管理器模块 226 和打印部 114 之间的借口。

并行端口驱动程序 228 提供 Web 服务器模块 203 的参照打印功能、Web 浏览器模块 211 的 Web 共享打印功能等数据打印功能通过并行端口向未图示的输出仪器输出数据时的 I/F (接口)。

图 3 是图象处理装置 110 的详细硬件结构的框图。

控制部件 111 与图象输入设备即读出部 113、图象输出设备即打印部 114 连接, 通过与 LAN 或公共线路 (WAN) 连接, 进行图象信息或设备信息的输入输出。

CPU301 是控制图象处理装置 110 的装置全体的控制器。

RAM302 是 CPU301 为了工作而使用的系统工作存储器。另外, RAM302 是用于暂时存储图象数据的图象存储器。

ROM303 是引导 ROM, 存储系统的引导程序。

HDD304 是硬盘驱动器, 存储系统软件和图象数据。

操作部 I/F306 负责与操作部 (UI) 112 之间的接口, 对操作部 112 输出操作部 112 的显示器 (未图示) 中显示的图象数据。另外, 也具有把用户通过操作部 112 输入的信息传递给 CPU301 的作用。

网络 (Network) I/F308 负责与 LAN100 的连接, 通过 LAN100 进行信息的输入输出。

调制解调器 (MODEM) 309 负责与公共线路的连接, 通过公共线路进行信息的输入输出。

以上的设备 301—306、308 以及 309 配置在系统总线 307 上。

图象总线 I/F (Image Bus I/F) 305 是连接系统总线 307 和高速传送图象数据的图象总线 310, 并且变换图象数据的数据构造的总线桥。

图象总线 310 由 PCI (Peripheral Component Interconnect) 总线或 IEEE1394 构成。

在图象总线 310 上配置以下的设备 311—316。

光彩图形处理器 (RIP) 311 把从网络发送的 PDL (Page

Description Language) 代码展开为位图图形。另外, 在 **PDL** 以外, 也进行用于把 **PDF (Portable Document Format)** 数据和各种应用程序固有的应用程序数据展开为位图图形的直接打印的处理。

设备 **I/F** 部 **312** 连接图象输入输出设备即读出部 **113** 以及打印部 **114** 和控制部件 **111**, 进行图象数据的同步系统/非同步系统的变换。

扫描图象处理部 **313** 对输入的图象数据进行修正、加工、编辑。

打印图象处理部 **314** 对打印输出的图象数据进行打印机的修正、析像度变换等。

图象旋转部 **315** 进行图象数据的旋转。

图象压缩部 **316** 对于多值图象数据进行 **JPEG** 压缩展开处理, 对于双值图象数据进行 **JBIG (Joint Bi-level Image Experts Group)**、**MMR**、**MH (Modified Huffman)** 的压缩展开处理。

图 4 是表示图象处理装置 **110** 的外观的图。

图象输入设备即读出部 **113** 对成为原稿的纸上的图象照明, 通过扫描 **CCD** 行传感器 (未图示), 生成光彩图形数据。

用户把原稿用纸设置在原稿供给器 **405** 的托盘 **406** 中, 如果在操作部 **112** 中指示原稿图象的读取, 则所述图 3 的 **CPU301** 对读出部 **113** 提供指示, 按照它, 供给器 **405** 一张一张供给原稿用纸, 读出部 **113** 进行原稿图象的读取动作。

图象输出设备即打印部 **114** 把光彩图形数据在用纸上打印。作为打印形式, 有使用感光体磁鼓或感光体带的电子照相方式、从微小喷嘴阵列喷出墨水并且在用纸上直接打印图象的喷墨方式, 但是可以采用任意的方式。须指出的是, 按照来自 **CPU301** 的指示进行打印动作。

打印部 **114** 为了能选择不同的用纸尺寸或用纸方向, 具有多个供纸段, 具有与它对应的用纸盒 **401**、**402** 和 **403**。另外, 排纸盘 **404** 接受打印完毕的用纸。

图 5 是表示操作部 **112** 的详细外观的图。

LCD 显示部 **501** 具有粘贴触摸屏薄板的 **LCD502**, 显示系统的操作画面以及由软件实现的按键 (以下称作“软件按键”), 如果按下显

示的按键，就把表示按下的位置的位置信息传递给所述 CPU301。

开始键 505 在开始原稿图象的读取动作时使用。在开始键 505 的中央部设置由绿和红等 2 的 LED 构成的 LED 显示部 506，通过该颜色表示开始键 505 是否处于能使用的状态。

停止键 503 进行工作中的动作的停止动作。

ID 键 507 在输入用户 ID 时使用。

复位键 504 在把来自操作部 112 的设定初始化时使用。

图 6 是表示操作部 112 的详细功能结构的框图。

如上所述，操作部 112 通过操作部 I/F306 与系统总线 307 连接，在系统总线 307 上连接 CPU301、RAM302、ROM303 和 HDD304。CPU301 根据 ROM303 和 HDD304 中存储的控制程序统一控制与系统总线 307 上连接的各种总线的访问。另外，CPU301 从通过设备 I/F312 连接的读出部 113 读入输入信息，对通过设备 I/F312 连接的打印部 114 输出作为输出信息的图象信号。RAM302 是 CPU301 的主存储器，作为工作区起作用。

来自触摸屏 502 和硬件实现的各种按键 503、504、505 以及 507 的用户输入通过输入端口 601 传递给 CPU301。CPU301 根据用户输入的内容和控制程序生成显示画面数据，通过控制画面输出设备的输出端口 602 对 LCD 显示部 501 输出生成的显示画面数据。另外，按照必要控制 LED 显示部 506。

图 7 是表示用于说明嵌入图象处理装置 110 中的应用程序的行动的网络结构一例的框图。

在图 7 中 4300 是远程复制的接收一侧（打印一侧）复合仪器。4350 是接收从综合发送部模块 204 通过多地址电报发送的图象数据，打印的激光打印机（LBP）等打印仪器。4400 是远程打印的接收一侧（打印一侧）的设备。4450 是接收通过多地址电报发送的图象数据，存储的组件服务器。4500 和 4600 是接收通过多地址电报发送的双值图象数据，存储的图象数据库服务器。4550 是接收通过多地址电报发送的电子邮件，存储的邮件服务器。4650 是具有信息内容的 Web 服

务器。4700 是访问 Web 服务器等的 Web 浏览器。

UI 模块 201 的细节如上所述，所以在此说明地址本模块 202。

地址本模块 202 保存在图象处理装置 110 的非易失性存储器例如电池支持的存储器和硬盘等中，在其中记载着网络连接的仪器的特征。具体而言，在地址本模块 202 中包含以下列举的信息。

即 (1) 仪器的正式名和别名

(2) 仪器的网络地址

(3) 仪器能处理的网络协议

(4) 仪器能处理的文档格式

(5) 仪器能处理的压缩类型

(6) 仪器能处理的图形的析像度

(7) 当打印仪器时，能供纸的纸尺寸和供纸段的信息

(8) 在服务器时，能存储文档的文件夹名。

以下说明的各应用程序根据地址本模块 202 中记载的信息，能判别发送目标的特征。另外，地址本模块 202 能编辑，并且下载保存在网络内的服务器中的地址，使用。或者能直接参照。

由遥控复制扫描模块 209 执行的远程复制应用程序按照对能由地址本模块 202 识别的发送目标指定的仪器能处理的析像度信息，把由扫描仪读取的双值图象数据 MMR 压缩并且 TIFF (Tagged Image File Format) 化后，通过 SLM 模块 215 发送给网络上的复合仪器 4300。

由综合发送部模块 204 执行的多地址电报发送应用程序与远程复制应用程序不同，能用一次的图像扫描对多个发送目标发送图像。另外，该发送目标不局限于打印仪器，也能对服务器等直接发送。以下按各发送目标说明。

当从地址本模块 202 识别发送目标的网络打印协议是 LPD (Line Printer Daemon)，并且能处理公开的打印记述语言 (PDL) 时，同样按照从地址本模块 202 识别的发送目标的仪器的图象析像度读取图象，通过编码器模块 221 把图象自身压缩，再 PDL 化，通过 LPR 模块 213 向发送目标的打印仪器 4350 发送。

当发送目标的仪器能通过 SLM 通信，并且是服务器时，从地址本模块 202 识别服务器地址和服务器的文件夹指定，与远程复制应用程序同样，把由扫描仪读取的双值图象数据 MMR 压缩，并且 TIFF 化，通过 SLM 模块 215 存储到网络上的服务器 4550 或 4500 等的特定文件夹中。另外，发送目标的服务器当判断为能处理 JPEG 压缩的多值图像时，把由扫描仪读取的双值图象数据 MMR 压缩，并且 TIFF 化，通过 SLM 模块 215 存储到网络上的服务器 4600 等的特定文件夹中。

当发送目标的仪器是电子邮件服务器时，识别地址本模块 202 中记载的邮件地址，把由扫描仪读取的双值图象数据 MMR 压缩，并且 TIFF 化，通过 SMTP 模块 214 向电子邮件服务器 4550 发送。此后的发送由电子邮件服务器 4550 执行。

由 Web 浏览器模块 211 执行的 Web 共享打印应用程序打印 Web 服务器 4650 等 Web 站点的信息。

由 Web 服务器模块 203 执行的 Web 服务器应用程序把由 HTML 记述的信息通过 HTTP 模块 212 提供给 Web 浏览器 4700。

该管理信息通过综合发送部模块 204、远程复制扫描模块 209、遥控复制打印模块 210 和控制 API 模块 218 取得，通过 HTTP 模块 212、TCP/IP 通信模块 216 和网络驱动程序 217 对 Web 客户机通知。另外，Web 服务器模块 203 对 Web 客户机提供参照打印功能。在参照打印中，Web 客户机对于 Web 服务器模块 203 以 HTTP 发送作为 POST 要求的内容（或嵌入 GET 要求的路径的一部分中）的 URL 信息。Web 服务器模块 203 接收 URL 信息，从接着接收的 URL 的位置通过 HTTP 取得数据，原封不动地把取得的数据传递给控制 API 模块 218，进行打印。在能由 HTTP 取得的位置放置由各种 PDL 记述的打印机能打印的数据时，客户机只通过对 Web 服务器指示 URL，就能进行所需的打印（不发送数据的实体，只发送“参照”）。打印机除了具有由各种 PDL 记述的数据，还能直接打印 PDF 的数据或应用程序数据的直接打印功能，所以能高效打印分散配置在网络的 Web 服务器

中的各种数据。

图 8 是表示 Web 浏览器模块 211 的软件结构的框图。

协议处理部 801 通过 HTTP 模块 212 与其它网络节点之间确立连接, 进行通信。具体而言, 对于由 URL 记述的资源发行 HTTP 要求, 取得其响应。在该过程中, 进行遵循各种编码形式的通信数据的编码和译码。协议处理部 801 如果取得具有由 HTML 等标记语言记述的构造的文本数据、嵌入该构造中的图象、脚本和风格等数据, 协议处理部 801 就把该数据传递给内容服务器 802。而协议处理部 801 取得页全体不具有一个图象数据形式或构造的文本数据形式的内容时, 协议处理部 801 把该数据传递给表现器 808。按照它, 表现器 808 进行表现, 把表现结果在 UI 模块 201 上显示。

协议处理部 801 当取得由 Web 浏览器模块 211 无法处理的形式的数据时, 协议处理部 801 进行委托给后面描述的代替处理的判断, 把该数据传递给控制 API 模块 218。

内容服务器 802 从协议处理部 801 取得以 HTML、XML (Extensible Markup Language) 和 XHTML(Extensible Hypertext Markup Language)等表现形式表现的内容数据, 进行字句分析和语法分析, 生成分析树。

DOM 构筑部 803 从内容服务器 802 取得分析树, 进行与内容数据的构造对应的 DOM 的构筑。以往的 HTML 允许语法上的各种省略, 变化跨多个分支。在现实世界中运用的内容即使是完整形式, 也常常不妥当。因此, DOM 构筑部 803 与其它一般的 Web 浏览器同样, 尝试妥当的 DOM 的构筑。

DOM 处理部 804 把 DOM 构筑部 803 构筑的 DOM 作为表现对象群的嵌套关系的树构造, 在存储器上保持管理。Web 浏览器模块 211 的各种处理以该 DOM 为中心实现。

布局引擎 807 按照 DOM 处理部 804 保持的对象群的树构造, 循环地决定各对象的显示上的表现, 结果取得文书全体的布局。有时通过嵌入文书中的记述或从文书链接的其他文件中的记述, 以 CSS

(Cascading Style Sheet)等风格板形式明确指定各对象的显示上的表现。

风格板服务器 806 分析与内容的文书关联的风格板。

布局引擎 807 放映基于风格板服务器 806 的风格板的分析结果，决定文书的布局。

表现器 808 按照布局引擎 807 决定的文书的布局，生成用于在 LCD 显示部 501 上显示的 GUI (Graphical User Interface) 数据。生成的 GUI 数据通过 UI 模块 201 在 LCD 显示部 501 上显示。

事件处理部 809 接收用户对操作部 112 上的触摸屏 502 或各按键进行的操作的事件，进行与各事件对应的处理。事件处理部 809 从控制 API 模块 218 接收装置或作业等的状态转移事件，进行与各事件对应的处理。

在 DOM 处理部 804 管理的 DOM 的树构造中，按对象的各类和各对象实例登记与各种事件对应的事件句柄。

事件处理部 809 按照生成的事件，从 DOM 处理部 804 管理的对象群之中决定应该担当该事件的处理的对象，发送事件。被发送了事件的对象按照与该事件对应的事件句柄的算法，执行各种处理。在事件句柄的处理中有 DOM 处理部 804 保持的 DOM 的更新、对于布局引擎 807 的再描画指示、对于协议处理部 801 的 HTTP 要求发行的指示、基于控制 API 模块 218 的调用的图象处理装置功能的控制等。

脚本翻译程序 805 是解释 JavaScript 或 ECMA Script 等的脚本，执行的翻译程序。脚本嵌入文书中，或者记述在从文书链接的其他文件中，进行对 DOM 的操作等。内容的提供者能通过脚本把提供的文书的动态举动编程。

图 9 是表示 LCD 显示部 501 中显示的 Web 浏览器的画面结构的图。

在图 9 中，标签 901 用于进行 Web 浏览器功能和其他功能（复制、框住、发送和扩张）的画面的切换。

URL 输入区 902 是用户输入所需的资源的 URL 的区域。如果用

户指定该区，就显示用于进行文字输入的虚拟的全键盘（未图示）。用户通过配置在虚拟的全键盘上的模拟键盘的软件按键能输入所需的字符串。

OK 按钮 903 是确定输入的 URL 字符串的软件按键。如果确定 URL，则 Web 浏览器模块 211 发出用于取得该资源的 HTTP 要求。

进度条 904 表示基于 HTTP 要求响应的内容取得处理的进度状况。

内容显示区 905 是显示取得的资源的区域。

返回按钮 906 是软件按键，追溯内容显示的历史，用于重新显示现在显示的内容之前显示的内容。

前进按钮 907 是软件按键，在追溯内容显示的历史，显示时，用于进展到前进到现在显示的内容之后显示的内容的现实。

重新加载按钮 908 是软件按键，用于进行现在显示的内容的再取得和再显示。

中止按钮 909 是软件按键，用于中止执行中的内容取得处理。

状态区 910 是显示来自图象处理装置 110 的各种功能的信息的区域。即使正在显示 Web 浏览器画面，也能显示从读出部 113、打印部 114 或其他功能发送的催促用户的注意的信息。另外，同样也从 Web 浏览器功能进行信息的现实。在 Web 浏览器功能中，显示链接目标的 URL 字符串、内容的风格的字符串、由脚本指示的信息。

图 10 是表示基于 HTTP 的要求和响应的处理流程的一例的序列图。

客户机 1001 是发送 HTTP 要求，接收 HTTP 响应的软件，相当于图象处理装置 110 内置的 Web 浏览器、PC (Personal Computer)、PDA (Personal Digital Assistant)、移动电话等中运行的一般的 Web 浏览器、用与 Web 浏览器同样的方法访问 Web 服务器进行服务的利用或中继的各种软件。

服务器 1002 是接收 HTTP 要求，进行对应的处理，再发送 HTTP 响应的软件，相当于以服务器 155 上运行的软件为首的 HTTP 服务器。

在 HTTP 要求 1003 中，客户机 1001 把对所需的请求的 GET 要求发送给服务器 1002。资源一般由 URI(Uniform Resource Identifier) (特别是 URL) 形式指定。服务器 1002 取得或生成 GET 要求中指定的资源所对应的数据，通过 HTTP 响应 1004 发回。在此，当指定的资源是与静态的文件对应时，服务器 1002 从服务器 155 的文件系统读出相应的文件，取得数据。而当指定的资源与 CGI 程序或 Servlet 等处理对应时，服务器 1002 执行相应的处理。即进行服务器一侧处理。服务器一侧处理生成对于要求的响应，但是在该过程中，有时也伴随着为了实现给定的服务所必要的业务逻辑的执行、对后端的数据库管理系统的访问等副作用。而且，发回作为服务器一侧处理的结果生成的数据。例如，当指定用于显示图象处理装置 110 的耗材目录的资源时，执行用于电子商务的软件，从数据库中参照用纸、粉末、零件的最新价格和库存状况等的记录，把这些信息整形为 HTML 形式或 XML 形式，进行生成目录文书数据的处理。

当由 HTTP 响应 1004 取得的数据为可显示的形式时，客户机 1001 进行该内容的显示。取得的数据是 HTML 文书，如果在文书中嵌入对其他资源的参照信息(图象、嵌入对象、其他 HTML 文书的取得源 URI)，Web 浏览器就按必要重复用于取得参照的其他资源的 HTTP 要求的发送。如果取得的数据是 HTML 文书，用户只通过选择 Web 浏览器上显示的页中的超文本等嵌入链接信息的 anchor，就能重复新的资源的取得和显示。

HTTP 要求 1005 是 POST 要求。HTML 文书包含(FORM)要素，当在该方法中指定 POST 时，如果用户提交 FORM 要素，就发送对指定资源的 POST 要求。在 POST 要求中，从客户机 1001 对服务器 1002 发送数据(通过响应，也进行从客户机 1001 到服务器 1002 的数据的发送)。在 Web 浏览器上显示的 FORM 要素中把用户输入的信息编码，通过 POST 要求对服务器 1002 发送。服务器 1002 的指定资源接收从客户机 1001 发送的数据，进行处理，生成 HTTP 响应 1006，发回。

根据作为 HTTP 要求而从客户机 1001 发送给服务器 1002 的信息中包含的协议标题的 User-Agent 属性，客户机 1001 对于服务器 1002 能表示自己是哪种客户机。在 User-Agent 的值中记述用于识别 Web 浏览器的安装的固有的文字列。

作为 HTTP 响应从服务器 1001 发送给客户机 1002 的数据由协议标题的 Content-Type 属性记述其数据类型。具体而言，在 Content-Type 中，数据类型的描述符按如下记述。即当通过 HTTP 响应传送 HTML 文书时，象 Content-Type: text/html; charset=Shift_JIS 那样记述。另外，当通过 HTTP 响应传送 PDF 文书时，象 Content-Type: application/pdf 那样记述。

图 11 是表示服务器一侧处理和客户机一侧处理的流程一例的图。

在 HTTP 要求 1003 下，客户机 1001 把对所需的请求的 GET 要求对服务器 1002 发送。资源一般由 URI (URL) 形式指定。服务器 1002 取得或生成 GET 要求中指定的资源所对应的数据，通过 HTTP 响应 1004 发回。当由 HTTP 响应 1004 取得的数据为可显示形式时，客户机 1001 进行该内容的现实。如果取得的数据是 HTML 文书，Web 浏览器就一边进行文书的解释，一边通过 DOM 构筑部 803 构筑 DOM。当在文书中嵌入 JavaScript 等的脚本时，把嵌入的脚本加载道对应的脚本解释程序中，执行客户机一侧处理。在客户机一侧处理中，能按照脚本的内容执行各种处理，但是能进行把脚本生成的数据插入正在分析的文书中等动态的文书操作。与文书的 onload 事件呼应的的事件句柄加在到脚本解释程序中，登记时，取得构成文书的全部资源，在分析的定时执行事件句柄的代码。通过该执行也能操作文书。

通过操作触摸屏等用户输入部件，为文书或文书的要素生成事件，或脚本解释程序监视的定时器的到时等事件发生时，作为事件句柄加载，如果有登记的脚本的代码，就执行它。

与服务器一侧处理时不同，客户机一侧处理的执行由客户机 1001 一侧的脚本解释程序处理，并不一定把用户输入传递到服务器 1002 一侧。当由单纯的计算实现的处理时，如果用客户机一侧处理实现，

就能省略与服务器 1002 的要求和响应的通信时间,只这部分就能改善转向时间。另外,因为各客户机 1001 分担处理,所以有避免负载集中到服务器 1002 一侧的计算资源中的效果。应该特别提到的是,在处理的实现中与服务器 1002 的对话通信变为不要,所以对网络或服务器 1002 不发送装置用户要求隐私的信息,保留在装置中,就能进行处理。

在现实中常常适当组合客户机一侧处理和服务器一侧处理两者,安装服务。

图 12 是表示管理利用图象处理装置 110 的用户的用户账户的数据构造一例的图,在用户账户信息的表中,各行与能利用装置的各用户对应。

在图 12 中,“system”和“guest”是图象处理装置 110 在内部使用的虚拟的用户名,除此之外,与实际利用图象处理装置 110 的实际的用户个人对应。

“system”是装置系统为了识别是拥有者的资源而利用。“guest”是在进行即使不认证用户,也能利用一部分资源那样的运用时,为了表现未认证的用户而利用。各列分别具有以下的意思。

即用户名的列是用于识别用户的标识符。在图 12 中,由字符串表现,但是在内部,可以用与各用户对应的数值表现。

基本组的列是表现各用户所属的既定组的标识符。各用户除了既定的组,还能属于其他多个组。后面描述组。

密码的列是通过 what-you-know 型的用户认证来认证用户时使用的密码。当通过智能卡功能进行用户认证时,可以不使用该列的信息,也可以在为了存储与智能卡并用的 PIN (Personal Identification Number) 而使用。

电子邮件地址的列表表示各用户的电子邮件地址。

内线的列表表示各用户的内线电话号码。

此外为了登记各用户的信息可以追加各种列。例如,如果追加登记既定的功能、既定的框内文件夹、喜欢的设定的列,就能把它们作为各用户固有的信息存储。

图 13 是表示管理利用图象处理装置 110 的用户组的数据构造一例的图，各行与用户所属的各组对应。

组是表现用户的集合的对象，能定义组织的部门、职务和担当的任务等各种集合。用户能同时属于多个组。

组名的列是用于识别组的标识符。在图 13 中，由字符串表现，但是在内部可以用与各组对应的数值表现。

成员列表的列表表示定义属于组的用户的列表。在本实施例中，“system”是作为用户系统的基本组使用的虚拟的组。“guest”相当于不进行用户认证而允许装置的利用时的客人用户。“admin”是进行装置的管理的管理者的组。“user”是在装置中具有用户账户的全部用户的组。“rd”和“sales”是相当于组织的部门的组。“manager”是相当于组织的职位的组。“operator”是相当于进行装置内部的数据备份等专门的操作的基于管理者的角色的组。

图 14 是表示把构成图象处理装置 110 的各种资源抽象化表现的资源类的数据构造一例的模式图。

资源类构成装置对象模型的各种对象的设计图中的类的一个。装置对象模型的对象(和作为设计图的类)在内部具有几个特性(特征)。特性包含保持数据或其他对象的参照的属性(或称作属性、数据成员、变量等)、操作(或方法、成员函数)。

装置对象模型的对象和类捆绑在脚本解释程序支持的各种脚本语言体系中，根据各语言的语法和语义，能利用。在 JavaScript 中，装置对象模型的对象与 JavaScript 对象类型关联。装置对象模型的类与 JavaScript 的意思中的类关联。须指出的是，与其它脚本语言的多数同样，JavaScript 不是“带强类型”的语言，不具有 Java 或 C++ 那样的意义的类的概念。但是，通过定义相当于类的对象，利用该对象的构造器和标准对象，实质上实现类具有的多个概念。

资源类 1901 是相当于图象处理装置 110 处理的全部资源的类的超类。资源类 1901 能保持多个实例属性和类属性。

所有者 ID 是实例属性，保持拥有该资源的用户的标识符。

组 ID 是实例属性，保持具有该资源的所有权的组的标识符。

对所有者的读出允许是实例属性，表现所有者 ID 代表的用户对该资源是否具有进行读出操作的权限。本属性能采用“可”、“不可”或“未设定”的任意一个值。

对所有者的写入允许是实例属性，表现所有者 ID 代表的用户对该资源是否具有进行写入操作的权限。本属性能采用“可”、“不可”或“未设定”的任意一个值。

对所有者的执行允许是实例属性，表现所有者 ID 代表的用户对该资源是否具有进行执行操作的权限。本属性能采用“可”、“不可”或“未设定”的任意一个值。

对组的读出允许是实例属性，表现属于组 ID 代表的组的用户对该资源是否具有进行读出操作的权限。本属性能采用“可”、“不可”或“未设定”的任意一个值。

对组的写入允许是实例属性，表现属于所有者 ID 代表的组的用户对该资源是否具有进行写入操作的权限。本属性能采用“可”、“不可”或“未设定”的任意一个值。

对组的执行允许是实例属性，表现属于所有者 ID 代表的组的用户对该资源是否具有进行执行操作的权限。本属性能采用“可”、“不可”或“未设定”的任意一个值。

对他人的读出允许是实例属性，表现与用户 ID 表示的用户、属于组 ID 表示的组的用户都不相应的用户对该资源是否具有进行读出操作的权限。本属性能采用“可”、“不可”或“未设定”的任意一个值。

对他人的写入允许是实例属性，表现与用户 ID 表示的用户、属于组 ID 表示的组的用户都不相应的用户对该资源是否具有进行写入操作的权限。本属性能采用“可”、“不可”或“未设定”的任意一个值。

对他人的执行允许是实例属性，表现与用户 ID 表示的用户、属于组 ID 表示的组的用户都不相应的用户对该资源是否具有进行执行操作的权限。本属性能采用“可”、“不可”或“未设定”的任意一个值。

既定的所有者 ID 是类属性，保持拥有该资源的缺省的用户的标

标识符。

既定的组 ID 是类属性，保持具有该资源的所有权的缺省的组的标识符。

对所有者的既定读出允许是类属性，表现所有者 ID 代表的用户对该资源是否具有进行读出操作的权限。本属性能采用“可”、“不可”或“未设定”的任意一个值。

对所有者的既定写入允许是类属性，表现所有者 ID 代表的用户对该资源是否具有进行写入操作的权限。本属性能采用“可”、“不可”或“未设定”的任意一个值。

对所有者的既定执行允许是类属性，表现所有者 ID 代表的用户对该资源是否具有进行执行操作的权限。本属性能采用“可”、“不可”或“未设定”的任意一个值。

对组的既定读出允许是类属性，表现属于组 ID 代表的组的用户对该资源是否具有进行读出操作的权限。本属性能采用“可”、“不可”或“未设定”的任意一个值。

对组的既定写入允许是类属性，表现属于所有者 ID 代表的组的用户对该资源是否具有进行写入操作的权限。本属性能采用“可”、“不可”或“未设定”的任意一个值。

对组的既定执行允许是类属性，表现属于所有者 ID 代表的组的用户对该资源是否具有进行执行操作的权限。本属性能采用“可”、“不可”或“未设定”的任意一个值。

对他人的既定读出允许是类属性，表现与用户 ID 表示的用户、属于组 ID 表示的组的用户都不相应的用户对该资源是否具有进行读出操作的权限。本属性能采用“可”、“不可”或“未设定”的任意一个值。

对他人的既定写入允许是类属性，表现与用户 ID 表示的用户、属于组 ID 表示的组的用户都不相应的用户对该资源是否具有进行写入操作的权限。本属性能采用“可”、“不可”或“未设定”的任意一个值。

对他人的既定执行允许是类属性，表现与用户 ID 表示的用户、属于组 ID 表示的组的用户都不相应的用户对该资源是否具有进行执

行操作的权限。本属性能采用“可”、“不可”或“未设定”的任意一个值。

当 JavaScript 等的脚本处理构成装置对象模型的对象时，脚本的权限等于操作 Web 浏览器的有效用户的权限。只当脚本中存在对象的读出权限时，允许来自脚本的属性的读出，只当脚本中存在对象的写入权限时，允许来自脚本的属性的变更，只当脚本中存在对象的执行权限时，允许来自脚本的操作的调用。在以上说明的允许和既定的允许属性中，“读出”、“写入”、“执行”的意思是有时对继承本资源类的子类表现的各种资源适当追加定义该资源固有的语义学。

以上说明的允许的实例属性和既定的允许的属性分别对应。当从某资源的类生成实例时，把使用该类的既定的允许的属性生成的实例的对应的允许的实例属性初始化。或者当某资源的实例中未设定允许的实例属性时，可以使用该实例的类保持的既定的允许的属性。

在本实施例中，由所述允许属性群表现资源的访问权限，但是通过按照相当于资源的各类保持公开的访问控制列表（ACL），表现资源的访问权限。

图 15A、15B、15C 是表示代表构成图象处理装置 110 的各种资源间的继承关系的数据构造一例的类图。

图 15A、15B、15C 的各类在生成（实例化）对象时使用。由这些类生成的对象群保持在 DOM 处理部 804 内部，构成脚本翻译程序 805 访问的装置对象模型。

图象处理装置 110 停止时，类和类之间的关系连续为字节列，作为文件永久地存储在 HDD304 中。在图象处理装置 110 的起动时和运行中，由控制部件 111 的 CPU301 读入该文件，本类图中所示的类群和该实例的对象群分别作为保持相互关系的数据构造动态地在 RAM302 内生成，维持管理。

资源类 1901 是把系统内的各种资源抽象化的类，在对应于多用户的图象处理装置 110 的访问控制中使用。资源类至少具有所述图 14 中说明的属性，它们被以下说明的全部子类继承，成为全部子类公共具有的特性。

部件类 1902 是资源类 1901 的子类，是把构成图象处理装置 110 的物理部件抽象化的类。

操作部部件类 1903、扫描仪部件类 1904、匹配部件类 1905、原稿供纸部件类 1906、供纸部件类 1907 以及排纸部件类 1908 分别是部件类 1902 的子类。

外部接口部件类 1940 是部件类 1902 的子类，是把用于外部连接的接口部件抽象化的类。

图象处理系统网络部件类 1941、USB 部件类 1942 和通信部件类 1943 是外部接口部件类 1940 的子类。

读出部类 1971 是部件类 1902 的子类，与读出部 113 对应。

打印部类 1972 是部件类 1902 的子类，与打印部 114 对应。

装置类 1970 是部件类 1902 的子类，与图象处理装置 110 对应。

外部部件类 1948 是部件类 1902 的子类，是把通过外部接口部件连接在装置上的外部部件抽象化的类。

传输类 1945 是资源类 1901 的子类，是把图象处理装置 110 通过网络或通信线路与其他节点通信的各种通信线路抽象化的类。

网络端口类 1946 是传输类 1945 的子类，是把通过网络接口与其他节点的进程通信的通信路线抽象化的类。网络端口类 1946 的实例例如通过 TCP(Transmission Control Protocol)或 UDP(User Datagram Protocol) 等协议和各协议的端口号付与特征。

通信线路类 1947 是把通过传真等中使用的通信线路与其他装置通信的通信线路抽象化的类。通信线路类 1947 的实例由分配给电话交换线路网的线路的线路号码付与特征。

功能类 1909 是资源类 1901 的子类，是把图象处理系统提供的功能抽象化的类。

复制功能类 1910 是功能类 1909 的子类，表示把纸原稿复制到纸上的功能。黑白复制类 1911 和彩色复制类 1912 是复制功能类 1910 的子类。

发送功能类 1913 是功能类 1909 的子类，表示通过各种协议发送

数字文书数据的功能。传真发送类 1914、电子邮件发送类 1915、文件传输发送类 1916、文件共享发送类 1917 以及数据库输入发送类 1918 是发送功能类 1913 的子类,表示通过与各自对应的协议发送数字文书数据的功能。

框功能类 1919 是功能类 1909 的子类,表示把从原稿读取的图象数据或由 PDL 展开取得的文书数据存储到系统内的存储器中,进行存储的文书数据的阅览、打印、发送、管理等的功能。

框存储类 1950 是框功能类 1919 的子类,相当于框功能类中把文书数据存储到存储器中的功能。

框输出类 1951 是框功能类 1919 的子类,相当于框功能类中读出存储在存储器中的文书数据并且进行的打印或发送等功能。

安全打印类 1952 是框功能类 1919 的子类,相当于框功能类中不原封不动打印从客户机 PC 接收的打印数据,暂时存储到存储器中,在用户从操作部 112 进行明确的打印操作时打印的留置打印功能。

接收功能类 1922 是功能类 1909 的子类,表示通过与发送功能类 1913 同样的各种协议接收数字文书数据,进行存储、传送、打印的功能。

扫描功能类 1953 是功能类 1909 的子类,表示使用读出部 113 读取原稿图象,变为电子的文书数据的功能。

打印功能类 1954 是功能类 1909 的子类,表示通过打印部 114 打印输出从客户机 PC 收到的打印数据的功能。

LIPS 打印类 1955 是打印功能类 1954 的子类,表示解释并展开从客户机 PC 收到的、公知的 LIPS 语言的 PDL 数据,打印输出的功能。

PostScript 打印类 1956 是打印功能类 1954 的字类,表示解释并展开从客户机 PC 收到的公开的 PostScript 语言的 PDL 数据,打印输出的功能。

PCL 打印类 1957 是打印功能类 1954 的字类,表示解释并展开从客户机 PC 收到的公开的 PCL 语言的 PDL 数据,打印输出的功能。

作业类 1921 是资源类 1901 的子类，是把功能类 1909 所对应的实际作业抽象化的类。

作业列表类 1958(附图中存在错误)是能存储多个作业的实例的容纳部，是把正在执行的作业列表或中断中的作业列表等一定种类的作业的集合抽象化的类。

历史类 1922 是资源类 1901 的子类，是把包含关于作业类 1921 的对象的执行和结束的信息、关于伴随着系统的动作而产生的各种情况的信息的历史信息的记录抽象化的类。

历史列表类 1959 是能存储多个历史的实例的容纳部，是把作业历史或系统历史等一定种类的历史的集合抽象化的类。

框类 1923 是资源类 1901 的子类，是把由功能类 1909 处理的文件系统的管理单位抽象化的类。

文件夹类 1924 是资源类 1901 的子类，是把由功能类 1909 处理的文件系统中用于把多个文件分组的管理构造抽象化的类。

文书 1925 是资源类 1901 的子类，是把由功能类 1909 处理的文件系统中的各文件即各种数字文书抽象化的类。

地址本 1926 是资源类 1901 的子类，登记系统的用户和用户的组，此外在发送功能类 1913 和接收功能类 1920 等功能中，是把处理地址信息时使用的地址的列表抽象化的类。

接收传送规则 1927 是把接收功能类 1920 中传送接收的文书时的传送条件或传送目标的指定或举动记述等的规则群抽象化的类。

设定类 1928 是资源类 1901 的子类，是把控制系统的各种动作的动作参数的设定抽象化的类。用户设定 1929 和管理者设定 1930 是设定类 1928 的子类，分别表示面向最终用户的设定和面向管理者的设定。

密钥和电子证明 1960 是资源类 1901 的子类，表示图象处理装置 110 保持管理，加密处理、用户认证、客户机认证、用户认证等中使用的各种电子密钥或署名或证明书等数据。密钥和电子证明 1960 的实例一般在安全性上极度重要，此外也包含各用户应该拥有的信息，所

以必须严格控制访问。

图 16 是表示代表图象处理装置 110 处理的各种资源间的聚合 (aggregation) 关系的数据构造一例的类图。

图 16 的各类在生成对象 (实例化) 时使用。由这些类生成的对象群保持在 DOM 处理部 804 内部, 构成脚本翻译程序 805 访问的装置对象模型。

图象处理装置 110 停止时, 类和类之间的关系连续为字节列, 作为文件永久地存储在 HDD304 中。在图象处理装置 110 的起动时和运行中, 由控制部件 111 的 CPU301 读入该文件, 本类图中所示的类群和该实例的对象群分别作为保持相互关系的数据构造动态地在 RAM302 内生成, 维持管理。该数据构造表现各种资源间的全部-部分的聚合关系。在本实施例中, 为了减轻管理与图象处理装置 110 关联的各种资源群的负担, 定义资源间的聚合关系。承认部件和子部件 (部分部件)、容纳部和内容等物理以及概念上的任意包含关系, 如果该关系有利于资源管理, 就导入。

图 16 中的各类分别与图 15A、15B、15C 中的对应的类相同。对于对应的类付与相同的符号, 省略详细的说明。

装置类 1970 表示图象处理装置 110 的全体。读出部类 1971 是装置类 1970 的部分。原稿供纸部件类 1906 和扫描仪部件类 1904 是读出部类 1971 的部分。

操作部部件类 1903 是装置类 1970 的部分。

打印部类 1972 是装置类 1970 的部分。供纸部件类 1907、标记部件类 1905、排纸部件类 1908 是装置类 1970 的部分。

网络部件类 1941 是装置类 1970 的部分。网络端口类 1946 是网络部件类 1941 的部分。

通信部件类 1943 是装置类 1970 的部分。通信线路类 1947 是通信部件类 1943 的部分。

USB 部件类 1942 是装置类 1970 的部分。外部部件类 1948 是 USB 部件类 1942 的部分。

框类 1923 是装置类 1970 的部分。文件夹类 1924 是框类 1923 的部分。文书 1925 是文件夹类 1924 的部分。

作业列表类 1958 是装置类 1970 的部分。作业类 1921 是作业列表类 1958 的部分。

历史列表类 1959 是装置类 1970 的部分。历史类 1922 是历史列表类 1959 的部分。

图 17 是表示构成装置对象模型的实例的对象的一部分的图。

在图 17 中，各四边形表示对象即各类的实例。在各对象的附近付与的编号表示与该对相对应的类的符号。

构成装置对象模型的对象具有该类具有的特性即属性和操作。除了对象的类中定义的特性，抽象-具体关系的父类的特性也通过继承具有。例如 Device 类的对象（装置对象）具有对装置类 1970 定义的特性、抽象类即部件类 1902、抽象类即资源类 1901 的特性。另外，从资源类的另一抽象类即 DOM 的 Element 类和 Node 类（未图示）、位于 JavaScript 的继承关系的最上级的 Object 类（未图示）继承特性。各对象通过其属性参照其部分对象。例如装置对 device 具有属性 printer，该属性参照打印部 1972 的实例。

装置对象 device 是装置类 1970 的实例，具有以下特性。

属性 serial 表示序列号。属性 vender 表示销售商。属性 model 表示模型。以上是从部件类 1902 继承的属性的一部分。

属性 firstChild 参照位于全体-部分关系的子的最初的对象。以上是从 DOM 的 Node 类继承的属性的一部分。

属性 printer 参照位于该对象的全体-部分的子的打印部类 1972 的实例。属性 reader 参照位于该对象的全体-部分的子的读出部类 1971 的实例。属性 communication 参照位于该对象的全体-部分的子的通信部件类 1943 的实例。属性 log 参照位于该对象的全体-部分的子的历史列表类 1959 的实例。以上是由装置类 1970 定义的属性的一部分。

操作 diag 进行图象处理装置 110 的自己诊断。该操作 dialog 是从部件类 1902 继承的操作的一部分。

操作 `getNumberOfJobs` 发回在图象处理装置 110 全体中正在执行的作业数。操作 `getCurrentUser` 发回表示正在利用装置的用户对象。以上是由装置类 1970 定义的属性的一部分。

操作 `getElementById` 从位于该对象的全体-部分关系的子的对象群中检索具有用 `id` 指定的 ID 的对象，发回该参照。操作 `getElementsByTagName` 位于该对象的全体-部分关系的子的对象群中检索具有用 `name` 指定的名字的对象群，发回它们的参照的列表。以上是从 DOM 的 Node 类继承的操作一例。

打印部类 1972 的实例即对象具有以下特性。

属性 `serial`、`vender`、`model`、`version`、`firstChild` 与装置对象 `device` 同样。属性 `isOnline` 是打印部 114 为在线时成为真值的描述语。属性 `finisher` 参照位于该部件的全体-部分的子的排纸部件类 1908 的实例。属性 `paperSupply` 参照位于该部件的全体-部分的子的供纸部件类 1907 的实例。属性 `marking` 参照位于该部件的全体-部分的子的标记类 1905 的实例。以上的属性是由打印部类 1972 定义的属性的一部分。

操作 `diag`、`getElementById`、`getElementsByTagName` 与装置对象 `device` 同样。操作 `getNumberOfJobs` 发回在打印部 114 中正在执行的作业数。操作 `getTotalCounter` 发回在打印部 114 中打印的累计打印数。操作 `getToner` 发回打印部 114 保持的粉末剩余量。操作 `setOnline` 使打印部 114 变为在线状态。操作 `setOffline` 使打印部 114 变为脱线状态。

图 18 是表示嵌入 Web 浏览器中嵌入的 JavaScript 翻译程序作为脚本的动作环境提供的客户机一侧对象的全体-部分阶层构造的一例的图。

在客户机一侧的 JavaScript 对象的全体-部分阶层中，位于最上级的是 Window 对象。与显示浏览器的文档的窗口对应的对象是 Window 对象。客户机一侧的对象全部从 Window 对象参照。Window 对象是与语言处理系统的范围链的开始连接的全局对象。Window 对象具有属性 `document` 或属性 `location`。属性 `document` 参照与对应的

窗口关联的 Document 对象。属性 location 参照与对应的窗口关联的 location 对象。Window 对象还具有属性 frames[]。如果利用属性 frames[]，就能参照与构成原来的窗口的各框架对应的 Window 对象。例如，属性 document 参照现在的窗口的第二子框架的 Document 对象。从现在的窗口或其他 Window 对象参照的对象也能参照自己以外的对象。例如，如果利用 Document 对象的 forms[]，就能参照与文行书中显示的多个 HTML FORM 要素 (FORM) 对应的 Form 对象。因为在这些 FORM 要素 (FORM) 中只参照任意一个，所以记述为 self.document.forms[0]。

在本实施例的 Web 浏览器中，装置对象模型位于该对象阶层中。Window 对象的属性 device 参照装置对象模型的实例(图 17 的装置对象 device)。

说明按以上构成的图象处理装置 110 执行的控制动作。

图 19 和图 20 是表示通过 Web 浏览器从服务器取得的文书的一例的图。

同一图的文书是嵌入 JavaScript 的代码的 HTML 文书。当 Web 浏览器解释该文书时，把嵌入的脚本加载到脚本解释程序中，执行，作为结果，动态适应该 Web 浏览器嵌入的特定嵌入装置的文书在客户机一侧处理中生成，显示。例如图 21 和图 22 是动态适应的文书的 Web 浏览器上的显示例。一边与显示例对应，一边以文书中嵌入的脚本部分为中心说明文书的内容。

在图 19 的行 7 中，通过装置对象 device 的 getCurrentUser 操作取得正在利用装置的用户对象，读出其 name 属性。结果把正在利用装置的用户名 (在显示例中，“山田太郎”) 插入文书中。

在行 9 中，根据装置对象 device 的 model 属性读出装置的模型名，把它 (在显示例中，“MF3200”) 插入文书中。

在行 12—行 33 中，对于构成装置的各子部件，按照它们的实际的安装状况，有选择地插入对于该固有的装置有益的信息。

在行 13 中，从装置对象 device 探索 reader 对象，读出 reader

对象的 `model` 属性，判定构成装置的读出部的模型是否等于“S500”。如果等于，在行 14 中就从 `reader` 对象探索 `feeder` 对象，读出其 `model` 属性，判定读出部 113 中安装的原稿供纸部件的模式是否等于“DF03”。如果等于，在行 15—行 21 中就把该特定的原稿供纸部件所固有的关联信息插入文书中。另外，在插入的信息中嵌入对更详细的信息源的链接，但是在行 19 和行 20 中，该链接的 URL 也动态生成最适合于由 `device.reader.model` 读出的原稿供纸部件的模型的 URL，向最佳的页引导。显示例的装置由该模型的读出部构成，并且安装有原稿供纸部件，所以插入介绍抵价促销的信息。

在行 24—行 32 中，记述构成装置的打印部的模型为“P9900”时的处理。显示例的装置不由该模型的打印部构成，所以在显示例中不插入该部分的信息。

在图 20 的行 35—行 56 中，比较服务器一侧把握的发布的固件的最新版本、嵌入实际的装置中的固件的版本，如果原封不动地嵌入老版本，就有选择地在文书中插入推荐基于客户工程师的升级作业的布置的信息。

在行 35 中，生成用于存储按各子部件发布的最近固件的版本的排列 `latests`。

在行 36—行 41 中，关于操作部、打印部、读出部、网络部件、通信部件和 USB 部件的各部件，生成存储表示最新版本的 `version` 属性和表示显示用的标签的 `label` 属性的对象，在排列 `latests` 中作为把对象名作为键的联想排列存储。

在行 42—行 55 中，构成关于排列 `latests` 的全部要素分别重复的循环。

在行 42 中，用变量 `unit` 指示排列 `latests` 的某要素的键。

在行 43 中，通过称作 `device[unit]` 的联想排列描述访问装置对象 `device` 的属性。例如如果 `unit` 的值为“`operationPanel`”，则 `device[unit]` 意味着 `device[“operationPanel”]`。在 JavaScript 中，它与用点描述的 `device.OperationPanel` 是相同的意思。因此，意味着对于装置对象

device 的 operationPanel 属性的访问。这是构成装置对象的操作部件对象。在该行中，判定 device[unit] 即某部件的对象在该系统的实际装置对象模型中，既不是未定义，也不是 null 值，并且 device[unit].version 即该部件的对象的版本是否为 latests[unit].version，即比行 36—行 41 中存储在排列中的该部件用固件的最新版本还小的值。当满足该条件时，意味着该部件的工作中的固件比发布的最近版老，此时在行 44—行 54 中，在文书中插入催促基于客户工程师的固件的升级（ROM 的更换等）的申请的信息。在插入的信息中嵌入参照基于 mailto: 疑似协议的 URL 的 anchor。在图 21 的显示中，该 anchor 作为“用 email 对支持中心申请”的超文本显示。用户如果点击该超文本，就从 Web 浏览器调用电子邮件客户机功能。在 URL 上，除了目标地址 support@canon.jp，还把 subject 和 body 编码，所以把目标地址、题名和本文的初始值传递给电子邮件客户机功能。

图 23 是这样起动的电子邮件客户机功能的画面例。在图示例中，设定内容提供者指定的目标地址。另外，设定包含从实际的装置对象取得的模型名和装置的序列号的题名。关于更新版本的固件存在的子部件，在本文中设定从实际的装置对象取得的子部件的模型名和现在嵌入该部件中的固件的版本。

回到图 20，在行 59—行 67 中，确认工作中的嵌入装置在该时刻的耗材剩余量，如果剩余量少，就催促耗材的订货，在文书中插入用于显示对在线订货页的链接的数据。

在行 59 中，通过操作 device.printer.getToner() 调用构成装置对象的打印部对象的 getToner() 方法，取得粉末剩余量的比例。

在行 60 中，判别粉末剩余量是否低于 10%，如果低于 10%，就进行对文书的插入。

在行 61 中，生成插入文书中的 URL，但是在此读出属性 device.printer.model，嵌入 URL 中，就能生成与适合于用户利用的模型的粉末的在线销售页直接链接的 URL。

在行 62 和行 63 中，催促注意粉末剩余量少，在文书中插入包含向时刻的粉末的在线销售页引导的超文本的信息。

当粉末剩余量为 10%以上时，如图 21 的显示例所示，在文书中不包含该信息，当低于 10%时，如图 22 的显示例所示，在文书中插入该信息。

在行 71 中，生成向服务器一侧提供的最新手册的链接。读出 device.model 属性的值，生成嵌入它的 URL，参照，所以如果用户探索该链接，就能直接到达适合于正在利用的装置的手册。

图 21 是动态适应的文书的 Web 浏览器上的显示例，表示 Web 浏览器的所述内容显示区 905 上的显示例。

在内容的文书中记述的信息 5001 中插入通过脚本从装置对象模型取得的正在利用图象处理装置 110 的用户名（山田太郎）。

在内容的文书中记述的信息 5002 中插入通过脚本从装置对象模型取得的装置的模式名（MFP3200）。

因为通过脚本从装置对象模型取得的判定读出部 113 的模型和原稿供纸部件的模型与信息提供的条件是否一致的结果为一致，所以段落 5003 插入文书中。另外，该段落 5003 中包含的 anchor 的 URL 包含从装置对象模型取得的原稿供纸部件的模型名，所以如果用户选择下划线的链接，就能直接转移到该部件的抵价促销页。

通过脚本比较内容中记述的子部件群的最新固件的版本和从装置对象模型取得的实际正在运行的子部件群的固件的版本，存在可升级的子部件，所以把段落 5004 插入文书中。另外，该段落 5004 中包含的链接包含用于邮件发送的 URL。在 URL 中，除了内容中记述的目标地址，还把从装置对象模型取得的装置的模式、序列号、升级作业对象的部件名、正在利用的固件版本的各信息编码，所以如果用户选择下划线的链接，就能自动生成对支持中心传递必要的信息的电子邮件。

链接 5005 是对放在服务器上的最新手册页的链接。在该 anchor 的 URL 中，把内容中记述的最新手册的路径和从装置对象模型取得

的装置模型名的信息编码，所以如果用户选择下划线的链接，就能直接转移到该嵌入装置专用的手册页。

图 22 是动态适应的文书的 Web 浏览器上的其他显示例，表示 Web 浏览器的内容显示区 905 上的显示例。该显示表示在显示图 21 的相同嵌入装置中，在不同的定时取得显示相同的内容的例子。

判定脚本调用装置对象模型的粉末剩余量取得操作取得的该时刻的粉末剩余量是否低于内容的脚本中定义的规定值（10%），结果为低于规定值，所以在文书中插入段落 5101。在该段落 5101 的信息中嵌入显示脚本调用装置对象模型的粉末剩余量取得操作取得的该时刻的粉末剩余量的值（8%）。

图 23 是按照由脚本处理动态生成的 URL 起动的电子邮件发送功能的画面例，邮件发送画面在所述操作部 112 中现实。

在图 23 中，在目标地址 5201 中设定内容提供者指定的目标地址。在题名 5202 中设定包含从实际的装置对象模型取得的模型名和装置的序列号的题名。在本文 5203 中，关于更新版本的固件存在的子部件，设定从实际的装置对象模型取得的模型名和现在嵌入该部件中的固件的版本。

自动设定布置客户工程师的出动所必要的信息，所以用户能原封不动发送电子邮件，如果希望，也能加写信息发送。

按钮 5204 是取消按钮，在不进行发送而关闭电子邮件发送画面时使用。

按钮 5205 是发送按钮，在进行电子邮件的发送时使用。

如果根据本实施例的图象处理装置，就能取得以下的效果。

即只通过在提供服务的服务器一侧准备单一的内容，就能提供按照系统的结构或状态有选择地提供适当的信息的文书。通过更新服务器一侧内容，能提供及时的信息，对服务器一侧内容的用户利用的固有的嵌入装置的适合能通过 JavaScript 的客户机一侧处理生成 HTML 的断片，插入内容文章中，执行，通过它们的组合，能灵活地提供极其细致的服务。例如能在与 Web 页内容的个性化、装置的结构和动态

的状态对应的最佳的提案中利用。另外，通过在动态生成的 URL 中反映装置的结构和状态，能用于显示从信息提示和在线销售等的 Web 页中直接转移到最适合于该装置的链接（此前，只能显示向排列并不一定最佳的选择分支的索引页的链接，要进行用户从其中选择最佳的选择分支的步骤）。另外，通过 URL 的生成，能在对支持中心发送的电子邮件的雏形中利用。这些全部对 CRM（Customer Relationship Management）有益。

下面说明应用本实施例 2 的嵌入装置的图象处理装置。

本实施例的图象处理装置与所述实施例 1 的图象处理装置相比，只是从服务器取得的文书的内容不同，所以包含本实施例的图象处理装置的系统的硬件结构和软件结构全部原封不动使用包含所述实施例 1 的图象处理装置的系统的硬件结构和软件结构。

图 24—图 26 是表示通过 Web 浏览器从服务器取得的文书的一例的图。

同一图的文书是嵌入 JavaScript 的代码的 HTML 文书。当 Web 浏览器解释该文书时，把嵌入的脚本加载到脚本解释程序中，执行，作为结果，动态适应该 Web 浏览器嵌入的特定嵌入装置的文书在客户机一侧处理中生成，显示。图 27 是动态适应的手册文书的 Web 浏览器上的显示例。一边与显示例对应，一边以文书中嵌入的脚本部分为中心说明文书的内容。

图 24 的行 18—行 26 中，判定选项部件即排纸部件的安装状况，如果安装，就显示用于直接显示关于该部件的手册的链接，如果未安装，就插入用于显示向购买排纸部件的页的链接的文书。根据装置对象模型 `device.printer.finisher` 属性中是否设置“null”以外的值（即参照 `FinisherUnit` 类的对象），判定排纸部件的安装状况。须指出的是，通过部件的该模型的图象（HTML 的 IMG 要素）显示 anchor 的内容（图 27 的区域 5301）。在未安装时，通过在半透明的部件上重叠“未安装”的文字的图象显示。

在行 31—行 38 中，同样判定选项部件即原稿供纸部件的安装状

况，如果安装，就显示用于直接显示关于该部件的手册的链接，如果未安装，就插入用于显示向购买原稿供纸部件的页的链接的文书。根据装置对象模型 `device.reader.feeder` 属性中是否设置“null”以外的值（即参照 `DocumentFeederUnit` 类的对象），判定原稿供纸部件的安装状况。须指出的是，通过部件的该模型的图象（HTML 的 IMG 要素）显示 anchor 的内容（图 27 的区域 5302）。在未安装时，通过在半透明的部件上重叠“未安装”的文字的图象显示。

在图 25 的行 43—行 52 中，同样判定选项部件即供纸部件的安装状况，如果安装，就显示用于直接显示关于该部件的手册的链接，如果未安装，就插入用于显示向购买供纸部件的页的链接的文书。根据装置对象模型 `device.printer.paperSupply` 属性中是否设置“null”以外的值（即参照 `PaperSupplyUnit` 类的对象），判定供纸部件的安装状况。须指出的是，通过部件的该模型的图象（HTML 的 IMG 要素）显示 anchor 的内容。在未安装时，通过在半透明的部件上重叠“未安装”的文字的图象显示（图 27 的区域 5305）。

在行 59 和行 60 中，显示用于直接显示与读出部 113 的模型对应的手册的链接。Anchor 的内容通过该部件的该模型的图象显示（图 27 的区域 5303）。

在行 67 和行 68 中，显示用于直接显示与打印部 114 的模型对应的手册的链接。Anchor 的内容通过该部件的该模型的图象显示（图 27 的区域 5304）。

在图 26 的行 78 和行 79 中，显示用于直接显示与读出部 113 的模型对应的手册的链接。通过包含部件的该模型名的分条写的文本，显示 Anchor 的内容（图 27 的文字列 5306）。

在行 83—行 86 中，判定选项部件即原稿供纸部件的安装状况，如果安装，就显示用于直接显示关于该部件的手册的链接。根据装置对象模型 `device.reader.feeder` 属性中是否设置“null”以外的值（即参照 `DocumentFeederUnit` 类的对象），判定原稿供纸部件的安装状况。须指出的是，通过包含部件的该模型名的分条写的文本，显示 Anchor

的内容（图 27 的文字列 5307）。

在行 90 和行 91 中，显示用于直接显示与打印部 114 的模型对应的手册的链接。通过包含部件的该模型名的分条写的文本，显示 Anchor 的内容（图 27 的文字列 5308）。

在行 95—行 99 中，判定选项部件即供纸部件的安装状况，如果安装，就显示用于直接显示关于该部件的手册的链接。通过包含部件的该模型名的分条写的文本，显示 Anchor 的内容。在图 27 的显示例中，未安装供纸部件，所以不进行链接的插入。

在行 102—行 106 中，判定选项部件即排纸部件的安装状况，如果安装，就显示用于直接显示关于该部件的手册的链接，如果未安装，就插入用于显示向购买排纸部件的页的链接的文书。根据装置对象模型 `device.printer.finisher` 属性中是否设置“null”以外的值（即参照 `FinisherUnit` 类的对象），判定排纸部件的安装状况。须指出的是，通过包含部件的该模型名的分条写的文本，显示 Anchor 的内容（图 27 的文字列 5309）。

图 27 是动态适应的手册文书的 Web 浏览器上的显示例，表示 Web 浏览器的内容显示区 905 上的显示例。图 27 的说明在图 23—图 26 的说明中一并进行，所以省略。

在本实施例 3 中，通过脚本的算法，按照部件的模型和选项部件的安装状况，动态切换文书的内容，但是进行与动态的状态相应的切换，按照装置中产生的错误，显示解说用于从该错误恢复的步骤的手册。另外，通过脚本的算法，从装置中存储的统计信息分析基于特定的装置的用户利用模式，显示最适合于该利用模式的“熟练使用”的引导。

根据本实施例的图象处理装置，能取得以下的效果。

即只通过在提供服务的服务器一侧准备单一的内容，就能提供按照构成系统的部件的模型或选项部件的安装状态，动态地提示适当的信息的文书。据此，例如能在提供通过嵌入浏览器浏览的最适合于该装置的状态的电子手册中利用。

下面说明本实施例 3 的应用嵌入装置的图象处理装置。

本实施例 3 的图象处理装置与所述实施例 1 的图象处理装置相比，只是从服务器取得的文书的内容不同，所以包含本实施例的图象处理装置的系统的硬件结构和软件结构全部原封不动使用包含所述实施例 1 的图象处理装置的系统的硬件结构和软件结构。

图 28 和图 29 是表示通过 Web 浏览器从服务器取得的文书的一例的图。

同一图的文书是嵌入 JavaScript 的代码的 HTML 文书。当 Web 浏览器解释该文书时，把嵌入的脚本加载到脚本解释程序中，执行，作为结果，动态适应该 Web 浏览器嵌入的特定嵌入装置的文书在客户机一侧处理中生成，显示。图 30 是动态适应的装置管理文书的 Web 浏览器上的显示例。

另外，在加载到脚本解释程序中的脚本中，包含通过对于 Web 浏览器上显示的文书要素的用户的输入操作而生成的 DOM 事件、与定时器事件呼应的事件句柄脚本，这些事件句柄在 Web 浏览器中显示文书后，按照各种事件执行。事件句柄通过对于装置对象模型的操作进行装置控制，此外通过对于 DOM 的操作，也进行取得 HTML 文书时解释、显示的文书的显示后的变更。

图 31 和图 32 是取得文书时的显示结束后新动态地变更的文书的 Web 浏览器上的显示例。一边与显示例对应，一边以文书中嵌入的脚本部分为中心说明文书的内容。

在图 28 的行 5—行 29 中，定义函数 `diag`。函数 `diag` 作为自变数，取得各部件对象的参照 `unit`，执行该部件的自诊断代码，根据执行结果进行已经在 Web 浏览器上显示的文书的编辑。

在行 6 中，调出在提供参照的部件对象中定义的自诊断方法即操作 `diag`（注意与行 5 中定义的 `diag(unit)` 不同）。然后把作为诊断结果发回的整数值作为 `diagResult` 取得。诊断结果如果是“0”，就正常，以外的数值意味着错误代码。

在行 7 中，通过 DOM 的操作，检索该 HTML 文书自身具有的带称作 `diagDiv` 的 ID 的 DIV 要素，取得参照 `diagDiv`。

在行 8 中，检索发现的 `diagDiv` 中包含的最初的 P 要素，取得参照 `oldP`。

在行 9 中，新生成 P 要素，取得参照 `p`。

在行 10 中，判定行 6 中执行的诊断的结果 `diagResult` 是否正常，如果正常，就执行行 11 和行 12，如果异常，就执行行 15—行 26。

当诊断代码正常结束时，在行 11 中生成由“诊断：正常。”的字符串构成的 DOM 的文本节点。

在行 12 中，把生成的文本节点作为行 9 中生成的 P 要素的最初的子追加。据此，在 `p` 的文本中设定“诊断：正常。”的字符串。然后，进入行 28。

而当诊断代码异常时，在行 15 中构筑嵌入诊断结果的错误代码 `diagResult` 的故障支持站点的 URL。

在行 16 中，生成包含诊断错误代码 `diagResult`，通告异常的信息的字符串构成的 DOM 文本节点。如果如果错误代码为“199”，则字符串变为“诊断：检测到异常。错误代码为#199。更详细的信息”

在行 19 中，把生成的文本节点作为行 9 中生成的 P 要素的最初的子追加。

在行 20 中，新生成 A 要素，取得它的参照 `a`。

在行 21 中，对生成的 A 要素即 `a` 设定 HREF 属性。HREF 属性的值是在行 15 中构筑的故障支持站点的 URL。

在行 22 中，生成由相同的 URL 字符串构成的 DOM 的文本节点。

在行 23 中，把生成的文本节点作为行 20 中生成的 A 要素 `a` 的最初的子追加。据此，Web 浏览器中显示的链接的用户能观察到的字符串成为故障支持站点的 URL。

在行 24 中，把设定完 HREF 属性和文本节点的 A 要素 `a` 作为 P 要素的第二子追加。

在行 25 中，生成由“请参照。”的字符串构成的 DOM 的文本节点。

在行 26 中，把生成的文本节点作为 P 要素的第三子追加。

在行 28 中,把 `diagDiv` 的最初的 P 要素更换为从此前显示的 `oldP` 新生成构筑的 `p`。

函数 `diag` 在图 29 的行 59—行 76 中记载的脚本中利用。

在行 59 中,确认 `device.printer.diag` 既不是未定义,也不是“`null`”,即确认登记有打印部的自诊断代码。如果登记,就在行 60 中,付与“打印机”的标签,作为点击事件的事件句柄,把登记以打印部对象作为自变数的函数 `diag` 的调用的 FORM 要素 (FORM) 的按钮插入 HTML 文书中。这样就显示用于调用打印部对象的自诊断的按钮。

同样,在行 62—行 65 中,如果提供自诊断代码,就显示用于调用排纸部件对象的自诊断的按钮。

同样,在行 66—行 69 中,如果提供自诊断代码,就显示用于调用供纸部件对象的自诊断的按钮。

同样,在行 70—行 71 中,如果提供自诊断代码,就显示用于调用读出部对象的自诊断的按钮。

同样,在行 73—行 76 中,如果提供自诊断代码,就显示用于调用原稿供纸部件对象的自诊断的按钮。

该脚本在取得 HTML 之后的解释时执行,所以按钮在 Web 浏览器上最初显示的页中显示。当在某特定的嵌入装置中提供打印部和读出部的诊断代码时,如图 30 的显示例那样,显示用于打印机和读出部的各按钮 5401 和 5402。

在用户点击 (因为是触摸屏,所以触摸) 显示的按钮的定时,通过作为事件句柄登记的函数 `diag` 执行诊断代码。例如,用户触摸写着“打印机”的按钮 5401,诊断结束,当检测到异常时,如图 31 的显示例的文字 5501 那样,变更正在显示的文书的一部分 (注意没理由再读入其他文书),标出对用户传递异常的信息。

在图 28 的行 32—行 41 中,定义函数 `showPrinterStatus`。

在行 33 和行 34 中,检索该文书内的带称作 `statusDiv` 的 ID 的 DIV 要素,取得该要素中包含的最初的 P 要素的参照 `p`。

在行 35 中,读出打印部对象的要素 `isOnline`,当 `isOnline` 的值

为“真”时，在行 36 中，对 p 的最初的子（即文本节点）的 data 属性设定“在线。”的字符串，当 isOnline 的值为“假”时，在行 39 中设定“脱线。”的字符串。

函数 showPrinterStatus()在图 29 的行 52 中利用。

在行 52 中，作为对于 BODY 要素的 onLoad 事件的事件句柄，登记脚本。在构成文书的全部数据的加载（展开、显示）结束的定时调用 onLoad 事件句柄。在句柄中，通过 setInterval，进行以一定间隔重复产生的定时器事件所对应的定时器事件句柄的登记。如果进行该登记，则以 10,000 毫秒间隔呼出 showPrinterStatus。

在 Web 浏览器显示文书之后的状态下，如图 30 的显示例所示，在 statusDiv 中显示作为当初的 HTML 文书记述的(行 81)字符串”“正在取得状态...”5403。

如果经过 10 秒左右，就象图 32 的显示例的字符串 5601 那样，对用户提示从装置对象模型取得的装置的状态，然后在 Web 浏览器中显示该文书时，以 10 秒间隔持续更新该时刻的状态显示。

在图 28 的行 44—行 48 中，定义函数 errorHandler。函数 errorHandler 接受自变数 event。

在行 45 和行 46 中，检索该文书内的带称作 statusDiv 的 ID 的 DIV 要素，取得该要素中包含的最初的 P 要素的参照 p。

在行 47 中，在 p 的最初的子（即文本节点）的 data 属性中设定连接作为自变数而接受的对象 event 的属性 unit 和“:”和对象 event 的属性 message 的字符串。

函数 errorHandler 从行 49 利用。为了持该装置对象的错误事件，可以在装置对象 device 的 onerror 属性中登记事件句柄函数。装置对象模型作为装置控制和借口，如果在装置控制中产生任意的错误，就调用 device.onerror 中登记的函数。此时，传递给自变数的 event 具有与错误关联的部件对象的参照 unit 或说明错误的字符串 message 等的属性。

图 30 是动态适应的装置管理文书的 Web 浏览器上的显示例，表

示 Web 浏览器的内容显示区 905 上的显示例。

图 31 是取得文书时的显示结束后新动态地变更的文书的 Web 浏览器上的显示例，表示 Web 浏览器的内容显示区 905 上的显示例。

图 32 是取得文书时的显示结束后新动态地变更的文书的 Web 浏览器上的其他显示例，表示 Web 浏览器的内容显示区 905 上的显示例。

根据本实施例的图象处理装置，能取得以下的效果。

即脚本通过操作 DOM，不仅分析文书时的插入，还能进行动态的文书的编辑。

通过装置对象模型的接口能从脚本控制装置或作业。

与 DOM 事件呼应，能登记操作装置对象模型的事件句柄。按照来自用户的输入和定时器等事件，能访问装置控制。

能与处理 DOM 事件同样对待来自装置对象模型的事件。

通过嵌入 HTML 文书中的脚本组合对于 DOM 和装置对象模型的操作，能通过 HTML 和脚本记述进行装置控制的用户界面。不仅在服务器一侧工作的 Web 应用程序的用户界面，在提供客户机一侧的装置服务的用户界面时也能利用作为 HTML 的用户界面记述语言的侧面。用户界面的记述作为服务器一侧的内容准备，按照来自装置的浏览器的取得要求取得，所以能在服务提供者一侧提供最新的内容。

据此，例如对于帮助基于客户工程师或用户的嵌入装置的维护是有效的。

下面说明应用本实施例 4 的嵌入装置的图象处理装置。

本实施例的图象处理装置与所述实施例 1 的图象处理装置相比，只是从服务器取得的文书的内容不同，所以包含本实施例的图象处理装置的系统的硬件结构和软件结构全部原封不动使用包含所述实施例 1 的图象处理装置的系统的硬件结构和软件结构。

图 33 和图 34 是表示通过 Web 浏览器从服务器取得的文书的一例的图。

同一图的文书是嵌入 JavaScript 的代码的 HTML 文书。当 Web 浏览器解释该文书时，把嵌入的脚本加载到脚本解释程序中，执行，

作为结果,动态适应该 Web 浏览器嵌入的特定嵌入装置的文书在客户机一侧处理中生成,显示。

另外,在加载到脚本解释程序中的脚本中,包含通过对于 Web 浏览器上显示的文书要素的用户的输入操作而生成的 DOM 事件、与定时器事件呼应的事件句柄脚本,这些事件句柄在 Web 浏览器中显示文书后,按照各种事件执行。事件句柄通过对于装置对象模型的操作进行装置控制,此外通过对于 DOM 的操作,也进行取得 HTML 文书时解释、显示的文书的显示后的变更。

图 35 是本文书的 Web 浏览器上的显示例。一边与显示例对应,一边以文书中嵌入的脚本部分为中心说明文书的内容。

在图 33 的行 5—行 7 中,定义函数 `createCounterReport()`。

在行 6 中,打印部 114 为了取得累计的打印总计计数器的现在值,调用 `device.printer.getTotalCounter()`,把它写入该文书中包含的 FORM 要素 (FORM) 的文本输入要素 `counter` 中。文本输入要素 `counter` 记述在图 34 的行 31 中。

在图 33 的行 8—行 10 中,定义函数 `reportLog()`。该函数接收自变量 `message`。

在行 9 中,把传递给函数的 `message` 追加到该文书中包含的 FORM 要素 (FORM) 的文本区域要素 `log` 中,改行。文本区域要素 `log` 记述在图 34 的行 36 中。

在图 33 的行 11—行 19 中定义函数 `createLogReport()`。

在行 12 中,清除文本区域要素 `log` 的内容。

在行 13 中,使用与 DOM 同样的 API 访问装置对象模型,从而检索历史列表类 1959 的对象。虽然历史列表对象能存在多个,但是在此把对第一历史列表对象的参照作为 `log` 取得。

在行 14 中,使用与 DOM 同样的 API 访问装置对象模型,从而从位于历史列表对象 `log` 的全体-部分关系的子的节点中检索标签名为“record”的对象群,把对该列表的参照作为 `records` 取得。

在行 15—行 21 中,构成循环,对于 `records` 中包含的各对象,

重复以下的处理。即在行 16 中，选择 records 中包含的一个对象，作为 record 参照，在行 17 和行 18 中，通过 DOM 的 API 即 record.getAttribute()，调查该记录对象具有的 DOM 属性。如果属性 facility 的值为“system”，并且属性 level 的值为“notice”，就在行 19 中，构筑把该对象记录的属性 date 的值和“:”和该对象记录的最初的子的 data 属性结合的字符串。然后传递在自变量中构筑的字符串，调用函数 reportLog()。

函数 createCounterReport()和 createLogReport()从图 34 的行 25 和行 42 的 HTML 利用。

在行 25 中，作为对于 BODY 要素的 onLoad 事件的事件句柄，登记脚本。onLoad 事件句柄在构成文书的全部数据的加载（展开、显示）结束的時刻调用。在句柄中，首先进行 createCounterReport()的调用，接着调用 createLogReport()。另外，在行 42 中，作为与带“更新”标签的按钮输入要素 updateButton 的 onClick 事件呼应的的事件句柄，登记脚本。与这些句柄完全同样，首先进行函数 createCounterReport()的调用，接着调用 createLogReport()。

在 Web 浏览器显示文书之后的状态下，打印部 114 的合计计数值输入到输入要素 counter（“计数器”）中，system 关联并且 notice 水平的历史记录对象的计时标记和信息输入到文本区域要素 log 中。因此，取得图 36 那样的显示。然后每当用户按“更新”按钮时，就通过该按钮的 onClick 事件句柄，更新计数器和历史的各区域的内容显示。

在该文书例中，在行 43 中，记述用于发送 HTML 的 FORM 要素（FORM）的 submit 型的输入要素。该输入要素在图 35 的显示例中与带“发送”标签的按钮对应。用户如果按“发送”按钮，则“计数器和“历史”的内容提交给服务器一侧。即能把脚本访问装置对象模型而取得的结果通过 FORM 要素（FORM）传递给服务器一侧。据此，能使服务器一侧处理和客户机一侧处理联合，组合对于 DOM 的操作和对于装置对象模型的操作的基于脚本的解决课题的机会增大。

但是，用户有时担心保持在嵌入装置内部的信息胡乱流出到外部

网络。存在即使用户不按提交按钮，也能从脚本生成与按下按钮相同的状态的方法。因此，根据嵌入装置的用户环境和运用形态，有可能更担心隐私的问题。

因此，脚本一度访问装置对象模型的任意的对象时，当指示来自该脚本嵌入的文书中包含的 FORM 要素 (FORM) 的信息的提交时，可以对用户确认是否真的执行发送处理。图 36 表示该确认对话框。来自 FORM 要素 (FORM) 的信息的提交指示是 HTML 文书的 DOM 事件，能由 DOM 管理部捕捉。另外，脚本访问的装置对象模型也嵌入 DOM 管理部，管理，所以 DOM 管理部能从文书中包含的脚本监视对装置对象模型的访问。DOM 管理部如果捕捉来自 FORM 要素 (FORM) 的信息的提交指示，就判定包含该 FORM 要素 (FORM) 的文书是否为不正常的和标记的，如果是不正常的，就中止提交指示的处理，对用户要求允许の確認。

图 35 是动态的装置报告文书的 Web 浏览器上的显示例，在 Web 浏览器的内容显示区 905 上显示图 33 和图 34 的文书。该文书的显示反映对装置对象模型的访问，动态地适应，此外具有起动对于装置对象模型的操作的用户界面 (“更新”按钮)、把结果反馈给 Web 浏览器的用户界面 (“计数器”和 “历史”区域)，通过来自 FORM 要素 (FORM) 的信息的提交 (“发送”按钮) 发送对装置对象模型的访问结果。

图 36 是表示从进行对装置对象模型的访问的文书尝试 FORM 要素 (FORM) 的提交时，为了对用户确认其允许而显示的确认弹出窗口的一例的图。

在本实施例中，当脚本一度访问装置对象模型的任意的对象时，对用户确认来自该脚本嵌入的文书中包含的 FORM 要素 (FORM) 的信息的提交的允许，但是当该限制过于苛刻时，也能有选择地址把装置对象模型的几个对象作为监视的对象。另外，也可以监视构成装置对象模型的对象具有的特性中特定的方法的调用和特定的属性的访问，允许来自不进行这些的文书的 FORM 要素 (FORM) 的提交 (即对用户不要求确认)。

另外，在本实施例中，在提交时进行允许的确认，但是也可以预先固定地设定允许或禁止。另外，可以预先设定基于文书的获得源 URL 或 FORM 要素 (FORM) 的发送目标 URL 的规则，根据该制约自动判断允许或禁止。

根据本实施例的图象处理装置，能取得以下的效果。

即能与处理 DOM 同样地处理装置对象模型。都是基于文本，此外进行一般的 Web 设计的制作者习惯于基于脚本的 DOM 的操作，所以能轻松制作与嵌入装置联合的内容。

能把装置对象模型的动态节点特别是历史的记录或多个可安装的同一种类的选项部件等需要构造上的灵活性的对象的集合作为与 DOM 同样的动态树构造探索和操作。

下面说明应用本实施例 5 的嵌入装置的图象处理装置。

包含本实施例的图象处理装置的图象处理系统的硬件结构和软件结构与包含所述实施例 1 的图象处理装置的图象处理系统的硬件结构和软件结构局部不同。以下只说明基于部分变更和追加的差异。

图 37 是表示 HDD304 的逻辑使用方法的图。

在本实施例中，按照使用用途把 HDD304 的存储区在逻辑上分为临时区 701 和框区 702。临时区 701 是改变图象数据的输出顺序，或在多部输出中能用一次的扫描输出，暂时存储 PDL 的展开数据和来自扫描仪的图象数据的存储区。框区 702 是用于使用框功能的存储区，象区域 703 那样分割为登记数量的小存储区。能对各框付与框名。用户通过指定框，就能把 PDL 或扫描作业作为各框内的文书存储。另外，也能生成用于汇总管理多个文书的文件夹。也能阅览框中存储的文书或文件夹，或打印、发送。

框 703—707 与其它装置内资源同样，被分配具有所有权的用户或组。另外，对于所有者即用户（以下称作“用户 u”）、属于所有的组的用户（以下称作“用户 g”）、不属于它们的其他用户（以下称作“用户 o”）等 3 种用户，分别作为读出、写入、执行的允许，能设定访问权。与文件夹以及文书同样，分配所有者和组，能设定访问权。

图 38 是作为资源的一例，表示参照和设定文书资源实例的访问全的用户界面的一例的图。

如果用户指定资源信息的参照或设定，就在 LCD 显示部 501 显示同一图的图形用户界面（GUI）。

在图 38 中，实例区表示现在选择的资源的实例。“技术调查报告书 0110”是该文书资源的实例带的标识符即文书名。

类区表示现在选择的资源的实例。“文书”表示该资源为文书类的实例。

阶层显示按钮是用于按阶层显示现在选择的资源的实例和类的关系的按钮。后面描述细节。

拥有者信息区是用于参照或设定关于现在选择的资源的拥有者的信息的区域。所有者区表示具有现在选择的资源的用户的标识符。组区表示具有现在选择的资源的组的标识符。

访问权设定区是用于参照或设定于现在选择的资源的访问权的设定的区域。本区域表示表，各行分别表示对该资源的实例的所有者的允许、对组的允许、对他人的允许的各状态。各列分别表示读出、写入、执行的各允许状态。行和列交叉的各区域表示该资源的分别对应的属性的值。

并不局限于该例子，关于构成装置对象模型的全部对象，只在具有读出权限时，允许来自脚本的属性的读出，只在有写入权限时，允许对脚本的属性的变更，只在有执行权限时，允许来自脚本的操作的调用。

各属性具有对各资源分别定义的意思。文书资源的读出允许表示能阅览、打印、发送该文书。文书资源的写入允许表示能编辑、改写或删除该文书。文书资源的执行允许表示把以该文书为对象的打印和发送等处理作为作业记述形式的作业脚本存储，当与该文书关联时，能执行作业脚本。

取消按钮是即使变更可变更的值，也使该变更无效，关闭本 GUI 的按钮。OK 按钮是变更可变更的值时，使该变更有效，关闭本 GUI

的按钮。

拥有该文书的用户通过使用本 GUI, 能从自己所属的组中设定拥有该文书的组, 此外对于所有者、组、他人分别能设定适当的允许。另外, 具有对管理者设定的写入权限的用户 (对未图示的管理者设定资源类的属性提供对于 admin 组的成员的写入权限) 能变更全部属性, 所以拥有该文书的所有者的设定也能变更。因此, 能进行所需的访问控制。在本实施例中, 所有者能进行对于该文书的全部操作, 对于所有者所属的组 rd (组织的研究开发部门) 的成员, 只允许读出操作, 对于此外的用户, 禁止包含读出的全部操作。

须指出的是, 在与文书资源关联的资源中存在文件夹类和框类。文件夹是把多个文书和多个其他文件夹作为内容容纳的容纳部, 框是顶层的文件夹。文件夹和框的允许表示中, 读出表示对容纳部中包含的内容的列表显示的允许, 写入表示对容纳部追加和删除内容的允许, 执行表示对各容纳部中可登记的内容的批处理的执行的允许。

图 39 是作为资源的其他一例, 表示参照和设定彩色复制功能资源实例的访问权的用户界面的一例的图。在图 39 中, 关于与图 38 同样的结构, 省略说明。

在图 39 中, 实例区表示现在选择的资源的实例。“<无名实例>”是对该彩色复制功能资源的实例付与的标识符即功能名。在本实施例中, 在彩色复制功能类的实例中只存在装置自动生成的单一的实例, 所以成为不具有特定的实例名的无名的实例。

类区表示现在选择的资源的类。“彩色复制”表示该资源为彩色复制类的实例。

阶层显示按钮是用于按阶层显示现在选择的资源的实例和类的关系的按钮。后面描述细节。

拥有者信息区是用于参照或设定关于现在选择的资源的拥有者的信息的区域。所有者区表示具有现在选择的资源的用户的标识符。组区表示具有现在选择的资源的组的标识符。

访问权设定区表示的各属性具有对各资源分别定义的意思。彩色

复制功能资源中的语义学由超类即功能资源定义。功能资源的读出允许表示能监视该功能的状态。功能资源的写入允许表示能变更以系统水平控制该功能的动作模式的特殊设定。功能资源的执行允许表示能起该功能的作业。

在图示的例子中，具有该功能的用户是 system，是在实际的利用者中不存在的虚拟用户。因此，具有对管理者设定的写入权限的用户通过使用本 GUI，能变更对于该功能的允许。组是 sales（组织的销售部门），只对属于该组的成员允许彩色复制功能的利用，对于此外的用户禁止。

图 40 是表示显示资源间的继承关系，参照和设定彩色复制功能类的访问权的用户界面的一例的图。

用户如果按下图 39 的阶层显示按钮，就在 LCD 显示部 501 上显示图 40 的 GUI。图 40 中，关于与图 39 同样的结构，省略说明。

在图 40 中，最上层的区域是阶层显示区。在阶层显示区中通过与图 15A、15B、15C 同样的阶层显示来显示基于抽象-具体关系的继承关系。在继承关系的显示中，各箱表现资源的类或实例。如果用户按下操作部 112 上任意的箱的区域，就能选择类或实例。由背景带斜线的矩形包围的类或实例表示现在选择的资源的类或实例。如果选择变更，则本 GUI 的以下的区域切换为关于选择的类或实例的显示。在图示例中，选择彩色复制功能类。

拥有者信息区是用于参照或设定关于现在选择的资源的拥有者的信息的区域。所有者区表示具有现在选择的资源的用户的标识符。组区表示具有现在选择的资源的组的标识符。

访问权设定区表示的各属性具有对各资源分别定义的意思。彩色复制功能资源中的语义学由超类即功能资源定义。功能资源的读出允许表示能监视该功能的状态。功能资源的写入允许表示能变更以系统水平控制该功能的动作模式的特殊设定。功能资源的执行允许表示能起该功能的作业。

在图示的例子中，具有该功能的用户是 system，是在实际的利用

者中不存在的虚拟用户。因此，具有对管理者设定的写入权限的用户通过使用本 GUI，能变更对于该功能的允许。组是 sales（组织的销售部门），只对属于该组的成员允许彩色复制功能的利用，对于此外的用户禁止。

图 41 是表示显示资源间的继承关系，参照和设定复制功能类的访问权的用户界面的一例的图。

用户如果按下图 40 的阶层显示区中表现复制功能类的箱，就在 LCD 显示部 501 中显示图 41 的 GUI。图 41 中，关于与图 40 同样的结构，省略说明。

在图 41 中，最上层的区域是阶层显示区。在图示例中，选择复制功能类。复制功能类是把彩色复制功能和黑白复制功能抽象化的超类，在继承子类中公共的却省属性值时使用。

在图示例中，具有该功能的用户是 system，是在实际的利用者中不存在的虚拟用户。因此，只有具有对管理者设定的写入权限的用户通过使用本 GUI，能变更对于该功能的允许。组是 user（全部用户），对全部用户允许复制功能的利用，对于此外的用户即 guest 用户禁止。

图 42 是表示显示资源间的继承关系，参照和设定黑白复制功能资源类的访问权的用户界面的一例的图。

用户如果按下图 41 的阶层显示区中表现黑白复制功能类的箱，就在 LCD 显示部 501 中显示图 42 的图形用户界面（GUI）。图 42 中，关于与图 41 同样的结构，省略说明。

在图 42 中，最上层的区域是阶层显示区。在图示例中，选择黑白复制功能类。黑白复制功能类是把复制功能类特殊化的类，从超类即复制功能类继承缺省的属性值。

在图示的例子中，具有该功能的用户是 system，是在实际的利用者中不存在的虚拟用户。因此，只有具有对管理者设定的写入权限的用户通过使用本 GUI，能变更对于该功能的允许。组是未设定，从超类的复制功能类继承 user（全部用户）。

在本访问权设定区中，未设定全部允许属性。因此，从超类的复

制功能类继承分别对应的允许属性值。

图 43 是表示显示资源间的聚合关系，参照和设定打印部资源类的访问权的用户界面的一例的图。

在 LCD 显示部 501 中显示图 43 的 GUI。图 43 中，关于与图 40 同样的结构，省略说明。

在图 43 中，最上层的区域是阶层显示区。在本阶层显示区中通过与图 16 同样的阶层显示来显示基于全体-部分关系的聚合关系。在聚合关系的显示中，各箱表现资源的类或实例。如果用户按下操作部上任意的箱的区域，就能选择类或实例。由背景带斜线的矩形包围的类或实例表示现在选择的资源的类或实例。如果选择变更，则本 GUI 的以下的区域切换为关于选择的类或实例的显示。在图示例中，选择打印部类。

拥有者信息区是用于参照或设定关于现在选择的资源的拥有者的信息的区域。所有者区表示具有现在选择的资源的用户的标识符。组区表示具有现在选择的资源的组的标识符。

访问权设定区表示的各属性具有对各资源分别定义的意思。本实施例的打印部类 1972 不过是为了把部分部件即供纸部件类 1907、标记部件类 1905 以及排纸部件类 1908 综合处理的部件，实际上不接受读出、写入或执行的任意访问。但是，打印部资源保持访问权设定属性，能参照和设定它。在本实施例中，把打印部的属性值作为该部分部件的既定属性值使用。因此，如果未设定位于全部-部分关系的子的多个部分部件的属性值，则这些部分部件的未设定属性值与全体部件的对应的属性联动。

图 44 是表示显示资源间的聚合关系，参照和设定排纸资源类的访问权的用户界面的一例的图。在图 44 中，关于与图 43 同样的结构，省略说明。

在图 44 中，选择排纸部件类。排纸部件类 1908 是打印部类 1972 的部分部件类。

在图 44 中，组合访问权的各属性未设定，所以处理为设定打印

部资源的对应的属性。

图 45 是表示违反访问权错误的显示一例的图。

当用户进行的处理由于访问权的不足而失败时，在 LCD 显示部 501 中显示图 45 的 GUI。GUI 的信息是用于对用户通知处理所必要的权限不足。另外，信息表示成为失败的原因的资源 and 对于该资源为必要的允许。如果用户按下细节按钮，就进一步详细显示关于错误的信息。作为细节显示，能显示成为失败原因的资源的信息（例如图 39）。用户如果按下 OK 按钮，本 GUI 就关闭。

在本实施例中，保持表现全体类和该部分类间的聚合关系的阶层构造，关于表示各资源的访问权的属性，全体类提供部分类的属性的既定值。因此，从全体类供给部分类中未设定的属性值，所以关于具有多个部分类的资源群，只通过用包含它们的类一度设定属性值，就能对各部分资源简易设定公共的缺省的访问权。另外，按照必要，对于更部分的资源（如果有，就是该部分类），能设定与公共的缺省值不同的特定的访问权。

具有按阶层显示各资源的类间的全体-部分的聚合关系的 GUI，具有一边参照类的聚合关系，一边参照和设定各类以及实例的访问权信息的 GUI。因此，用户容易理解把握各种资源间的聚合关系，根据把握的聚合关系，能容易进行适当的访问权的设定。

须指出的是，在所述各实施例中，DOM 只取出 HTML 的 DOM 进行说明，但是对于由 XHTML 等其他标记语言记述的文书也能应用本发明。例如通过从脚本操作 SVG 的 DOM，也能组合比只由 HTML 构成的文书更富于图形的用户界面和装置对象模型的操作。

另外，在所述各实施例中，使用 HTTP 说明浏览器和服务器的协议，但是 HTTPS(Hypertext Transfer Protocol Security)当然可以，如果是在包含今后的 Web 技术中一般使用的协议，就可以利用任意的协议。

在所述各实施例中，根据正在利用的用户决定伴随着脚本的执行的处理的权限，但是也可以不根据用户，唯一地决定脚本用的访问权，

或根据通过对脚本附加的署名等来识别的脚本提供者，决定访问权。

另外，在所述各实施例中，对各对象设定访问权，但是可以对类中包含的各操作设定权限。

以上是本发明的实施例的说明，但是本发明并不局限于这些实施例，如果是能实现权利要求书中表示的功能或实施例的结构具有的功能的结构，则无论是怎样的结构，都能应用。另外，可以应用于由多个仪器构成的系统中，也可以应用于由一个仪器构成的装置中。

须指出的是，通过对系统或装置供给记录了实现上述各实施例的功能的软件的程序代码的存储媒体，该系统或装置的计算机（或 CPU 和 MPU）读出存储在存储媒体中的程序代码，从而也能实现本发明的目的。

此时，从存储媒体读出的程序代码自身实现本发明的新功能，该程序代码以及存储该程序代码的存储媒体构成本发明。

另外，作为用于供给程序代码的存储媒体，例如能使用软盘、硬盘、光磁盘、CD-ROM、CD-R、CD-RW、DVD-ROM、DVD-RAM、DVD-RW、DVD+RW、磁带、非易失性存储卡、ROM 等。

另外，不仅通过执行计算机读出的程序代码，实现所述实施例的功能，还包括以下情况：根据程序代码的指示，在计算机上运行的 OS(操作系统)进行实际处理的一部分或全部，通过该处理，实现所述实施例的功能。

而且，还包括以下情况：把从存储媒体读出的程序代码写到插入计算机中的功能扩展板或连接在计算机上的功能扩展部件具有的存储器中后，根据程序代码的指示，功能扩展板或功能扩展部件具有的 CPU 进行实际处理的一部分或全部，并通过该处理来实现所述各实施例的功能。

图1

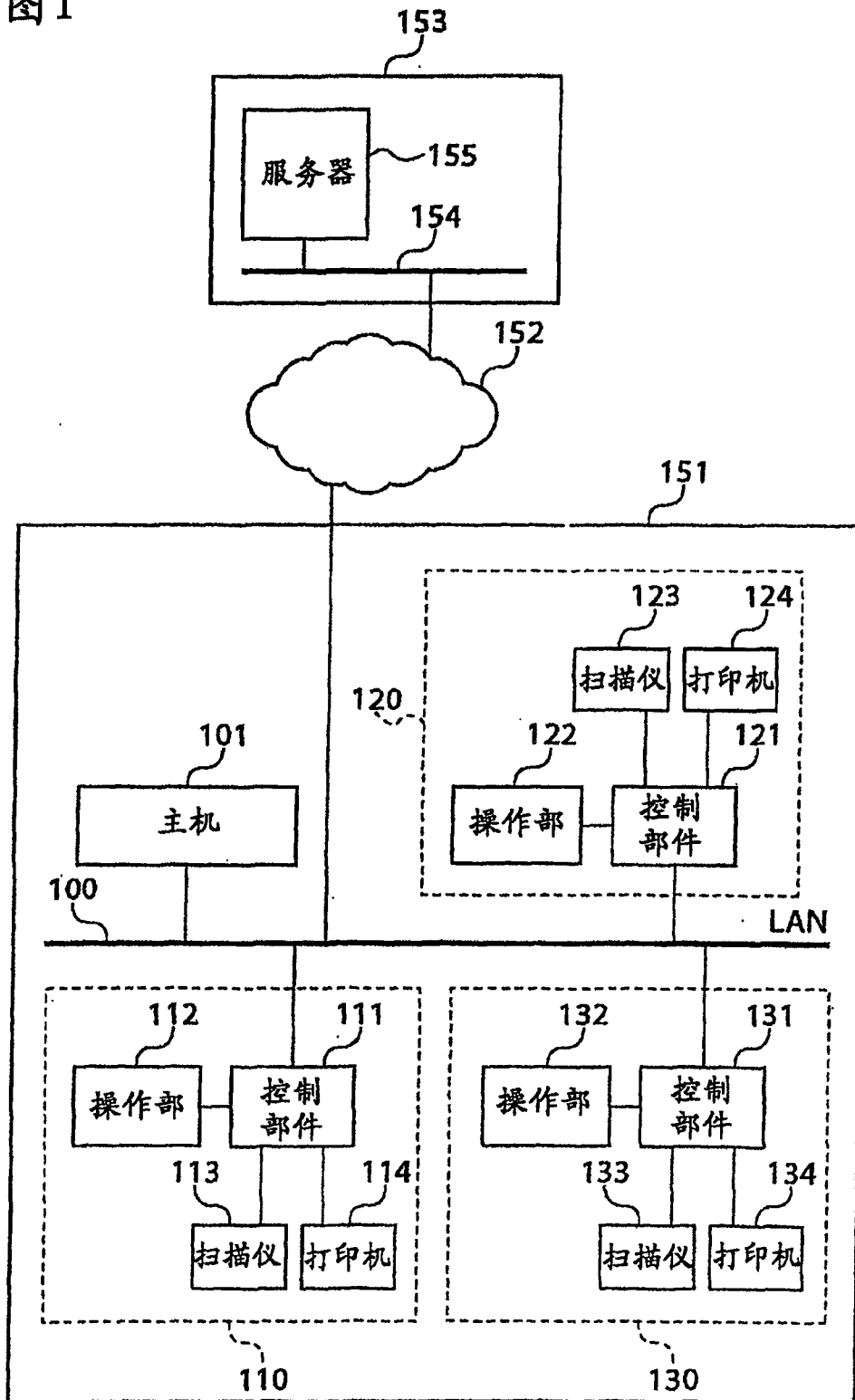


图2

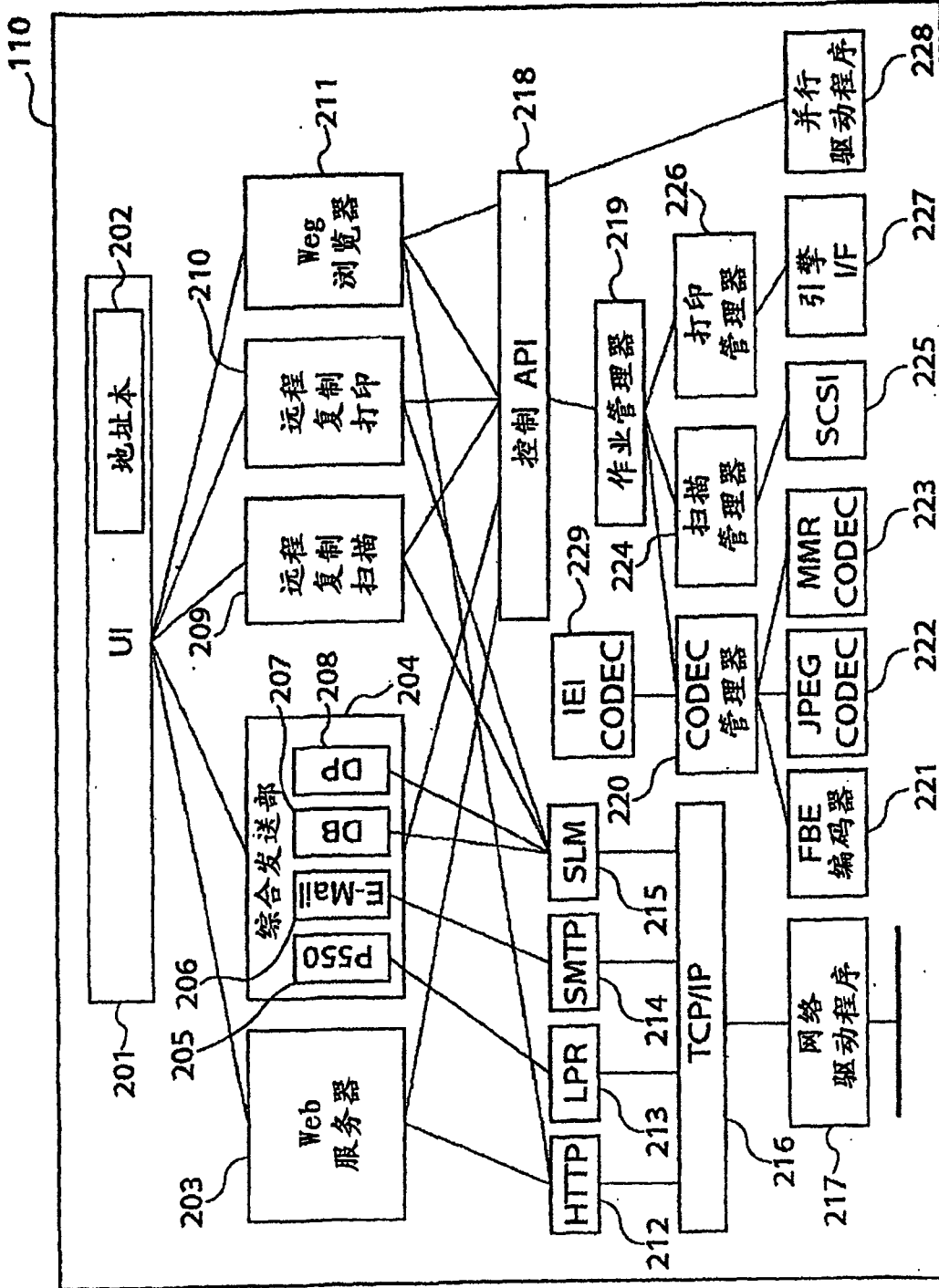


图3

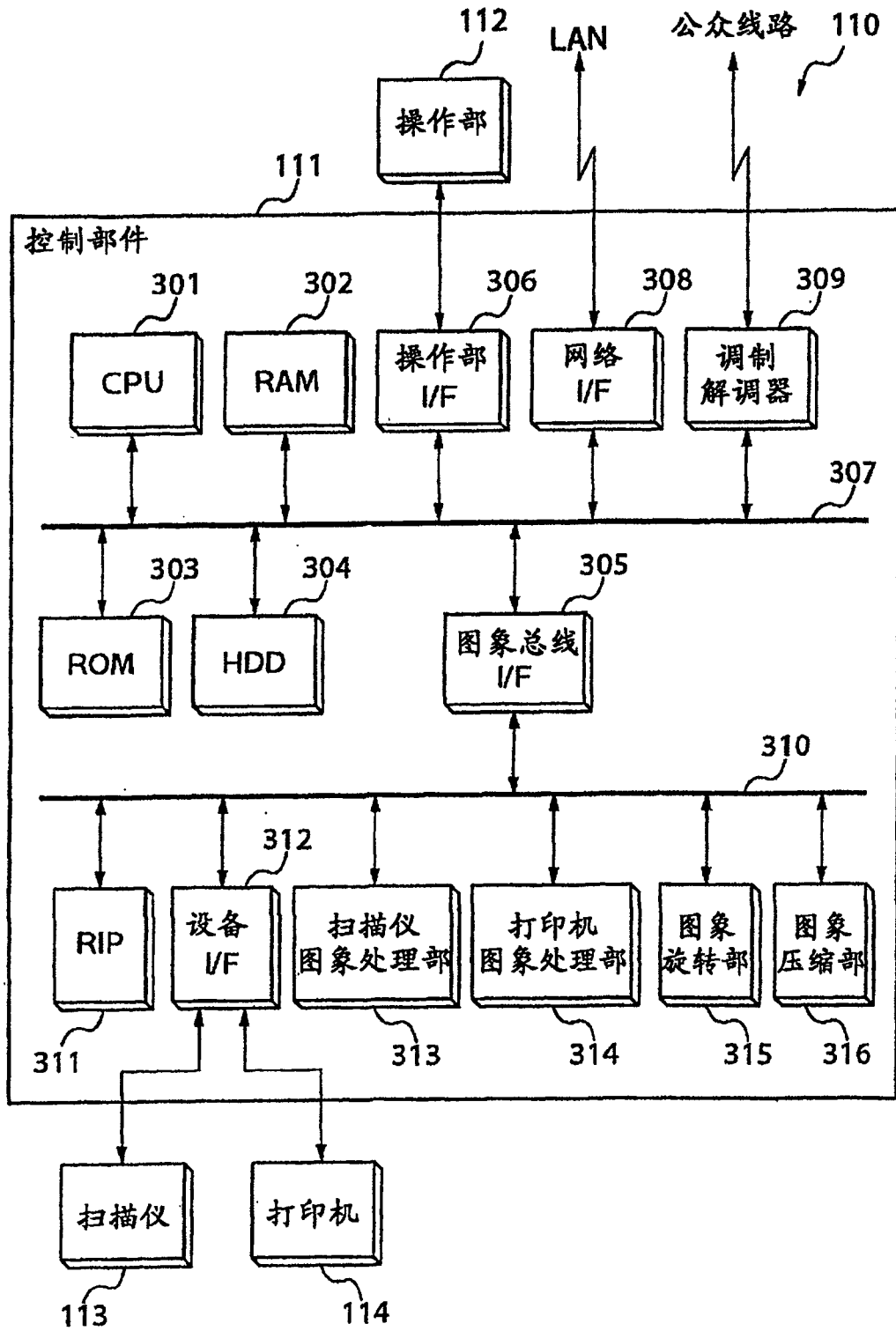


图4

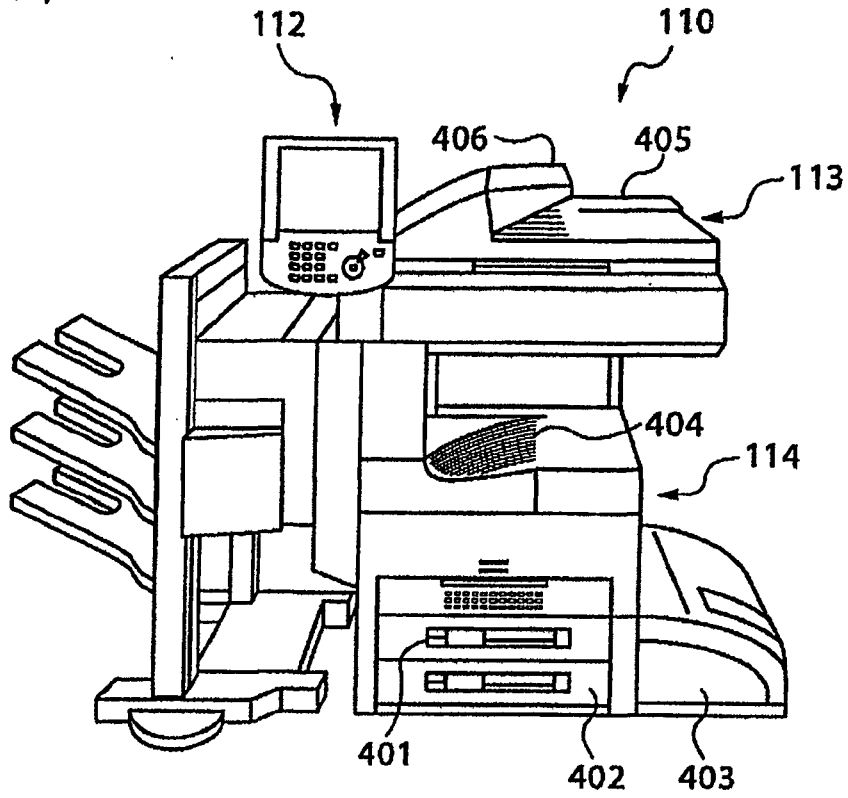


图5

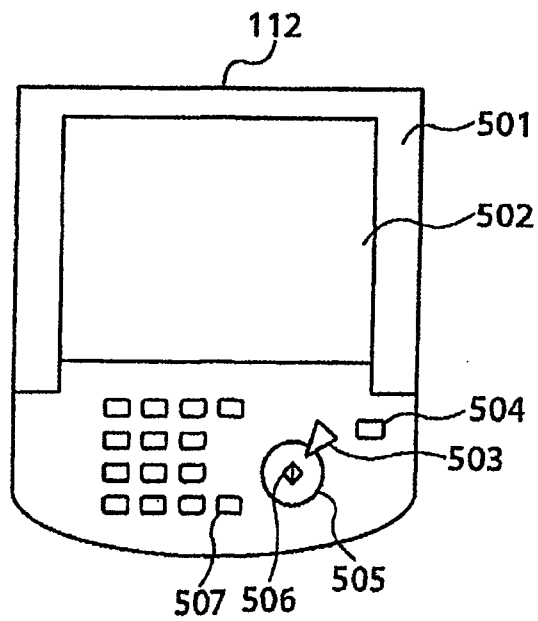


图6

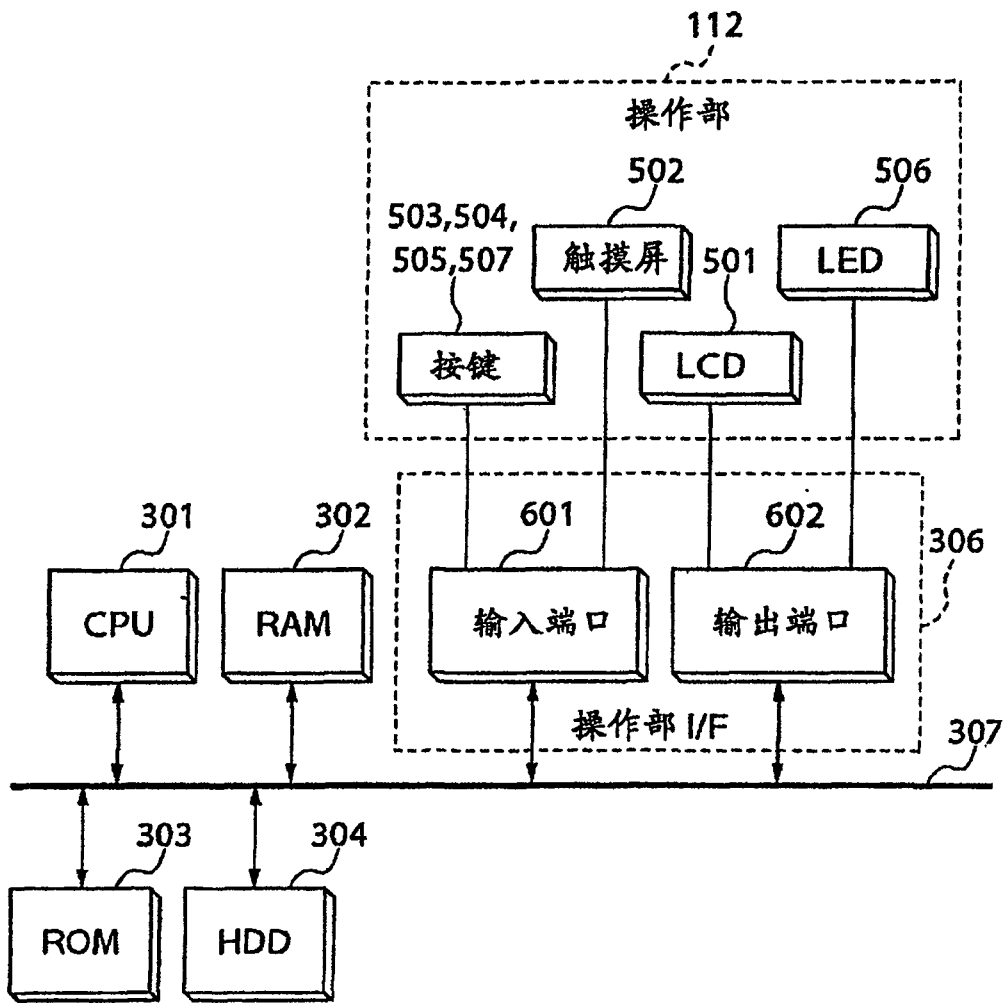


图7

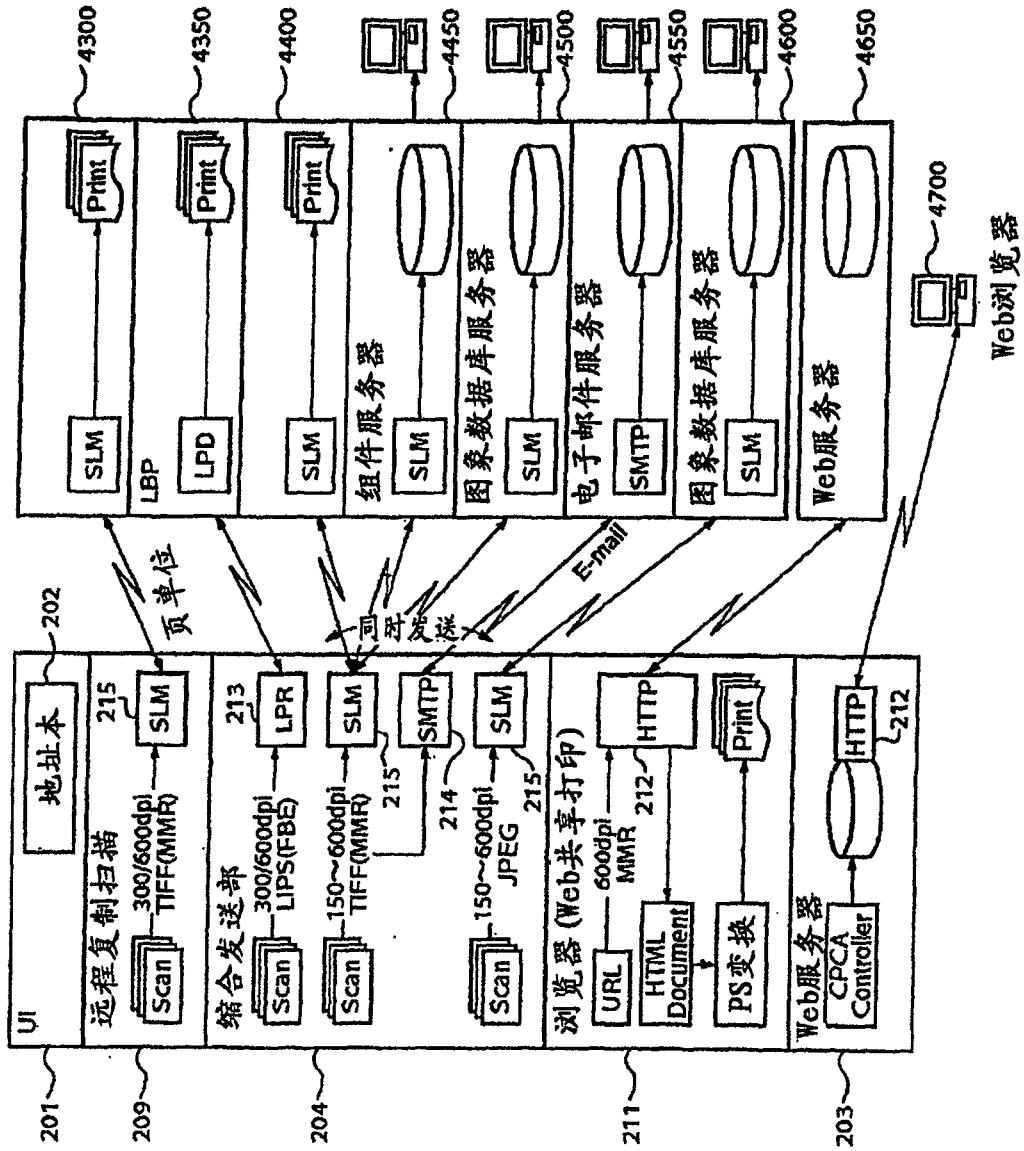


图8

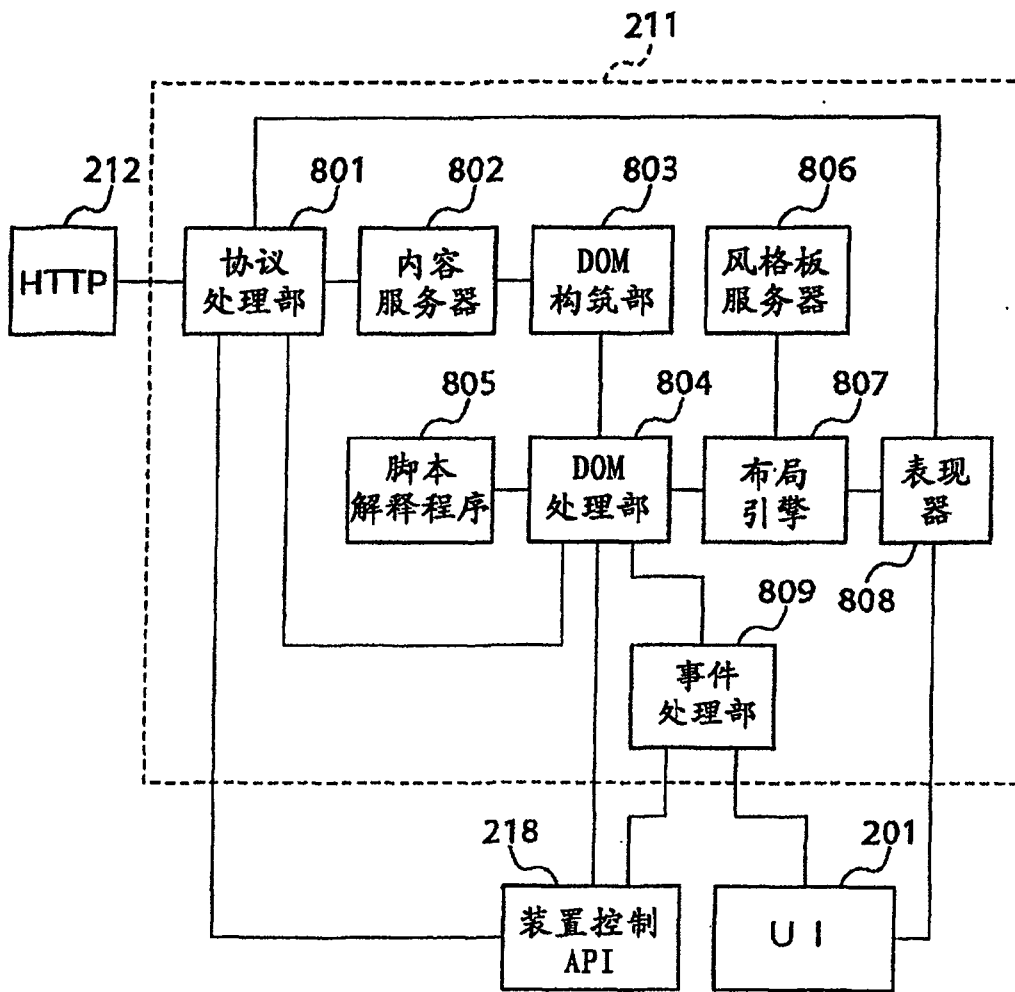


图9

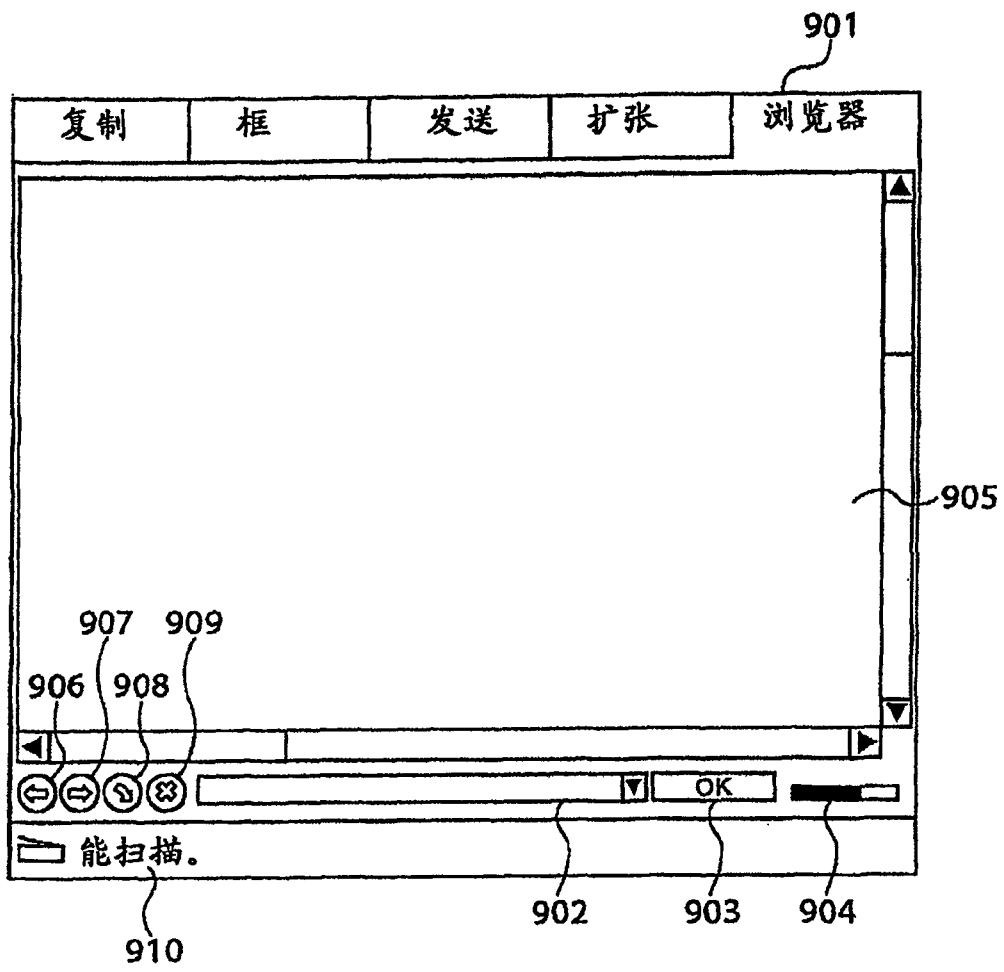


图10

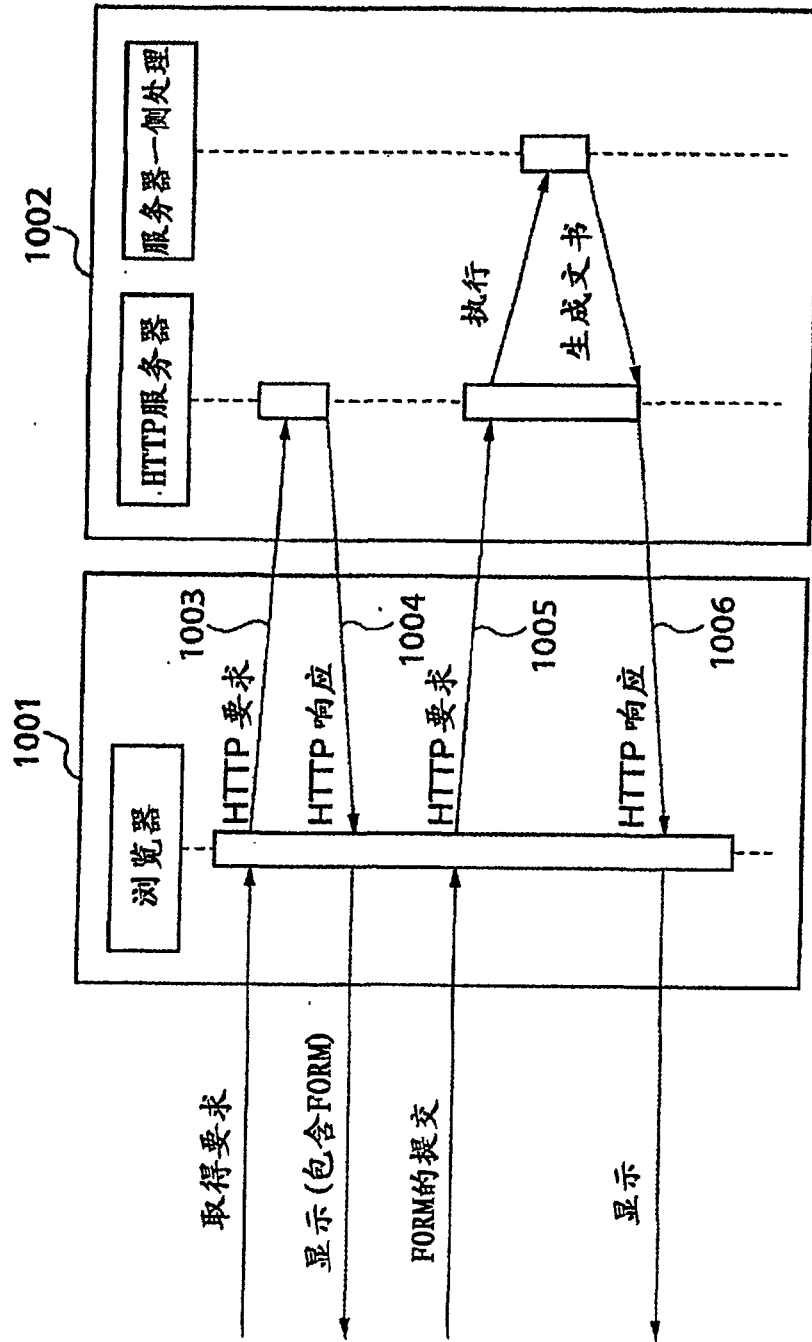


图11

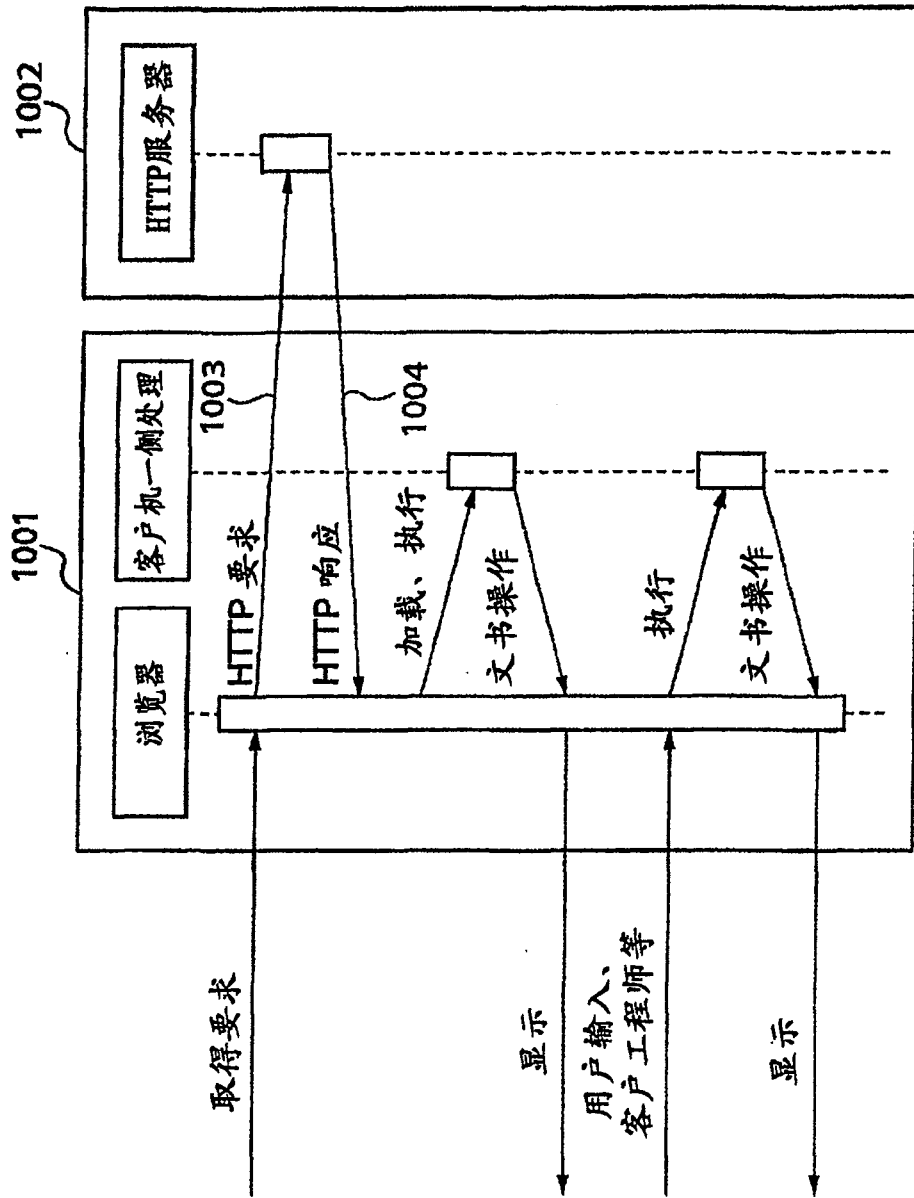


图12

用户名	基本组	名字	密码	电子邮件地址	内线	...
system	system	系统	*****			
guest	guest	客人				
kanazawa	rd	金泽文子	*****	kanazawa@do.ma.in	1230	
homma	rd	本真高司	*****	homma@do.ma.in	1231	
yam	rd	山田太郎	*****	yam@do.ma.in	1232	
ucchi	rd	...	*****	ucchi@do.ma.in	1234	
hikichi	rd	...	*****	hikichi@do.ma.in	1235	
foo	sales	...	*****	foo@do.ma.in	2200	
bar	sales	...	*****	sales@do.ma.in	2201	
...						

图13

组名	成员列表
system	system
guest	guest
admin	ucchi
user	kanazawa,homma,yam,ucchi,hikichi,foo,bar
rd	kanazawa,homma,yam,ucchi,hikichi
sales	foo,bar
manager	kanazawa,homma,foo
operator	ucchi,hikichi,bar
...	

图14

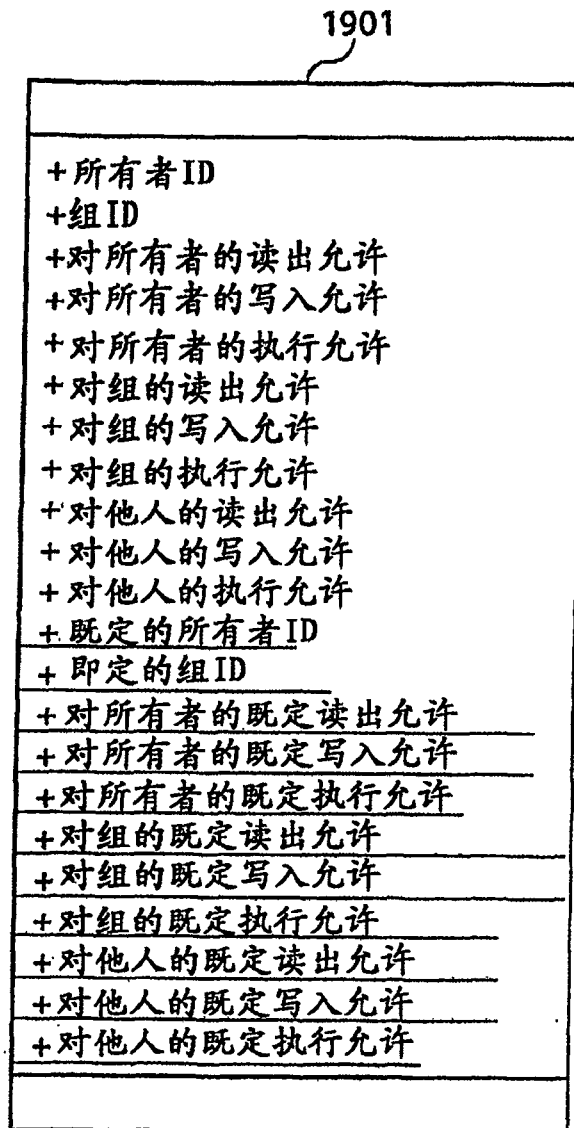


图 15A

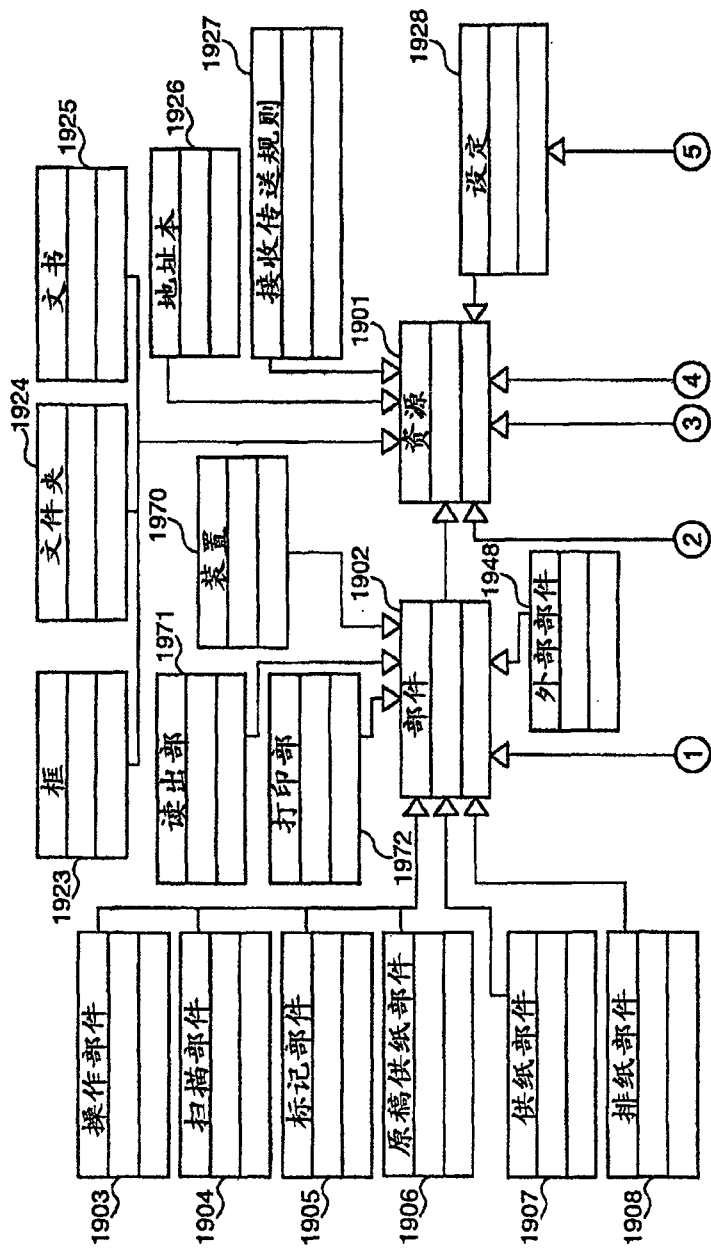


图 15B

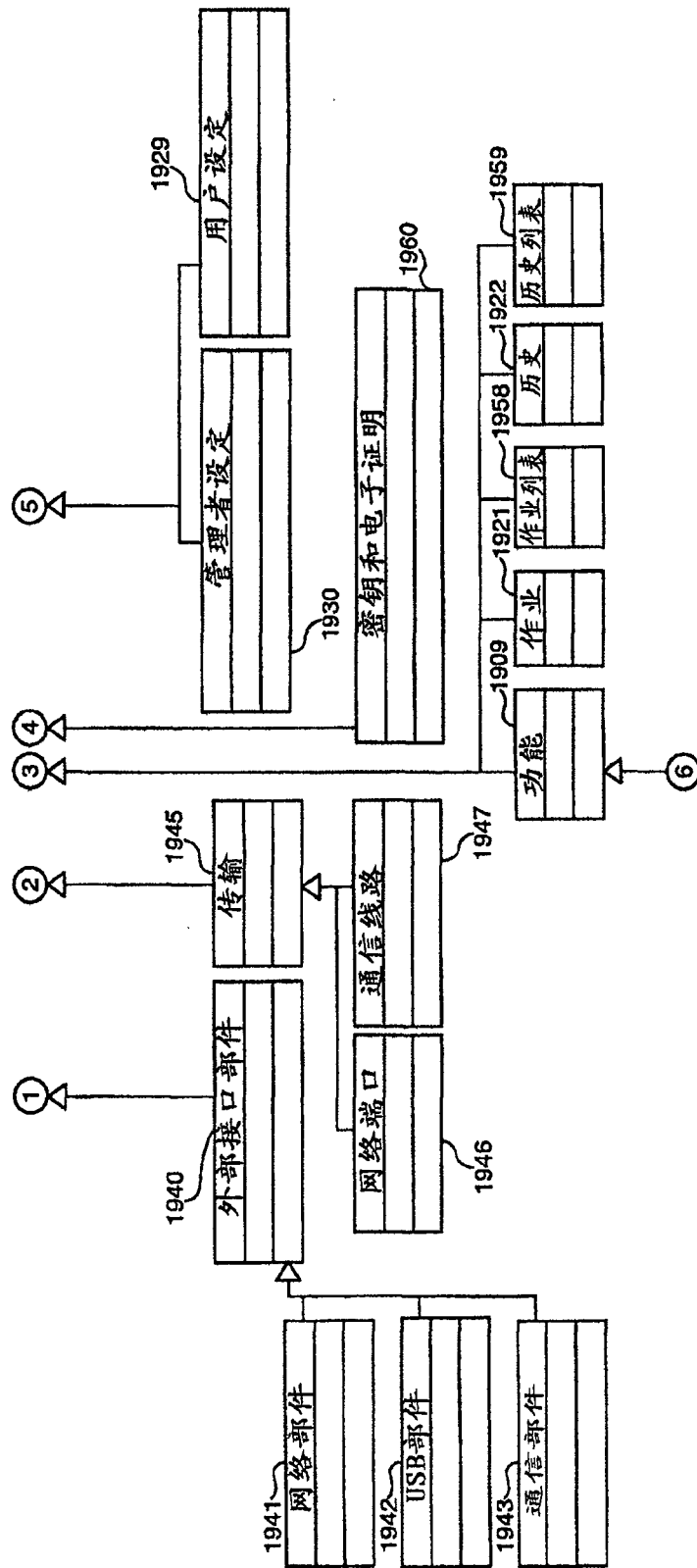


图 15C

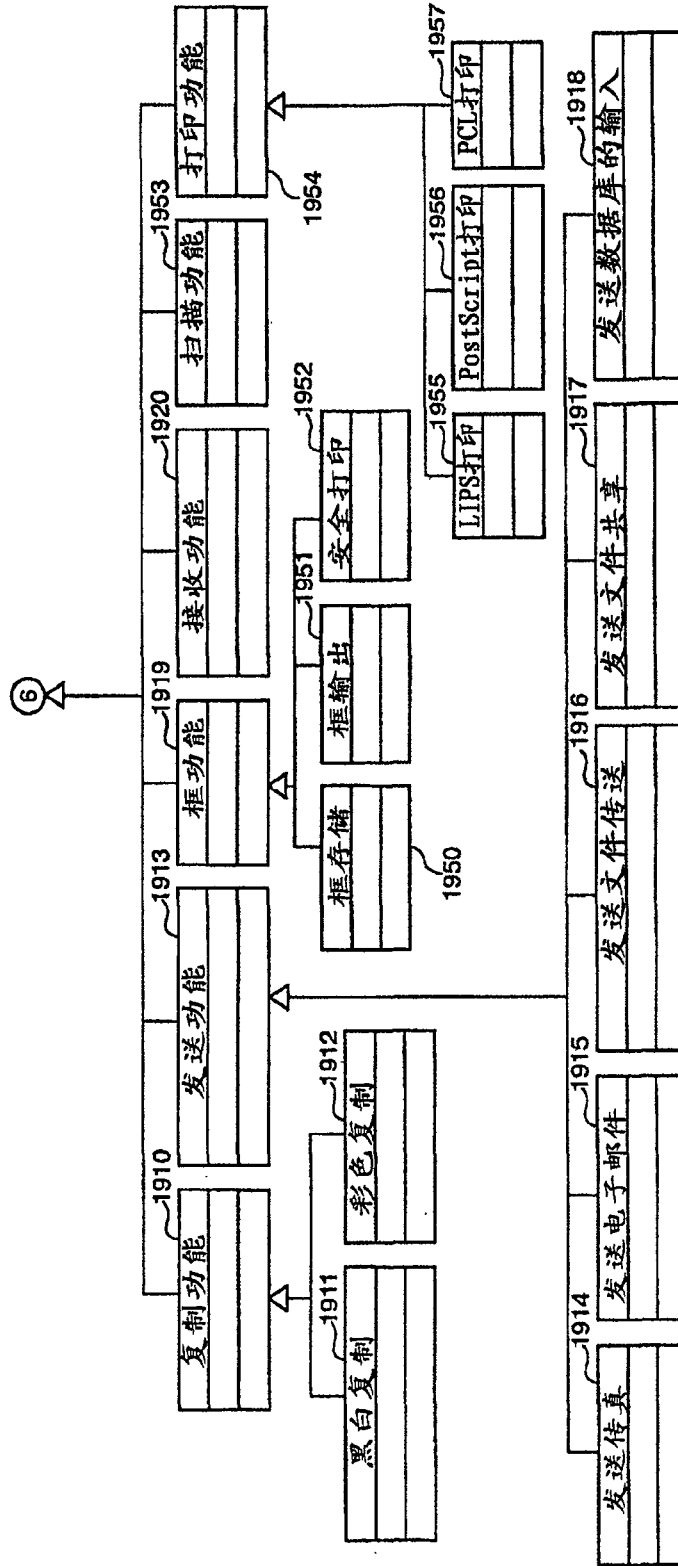


图16

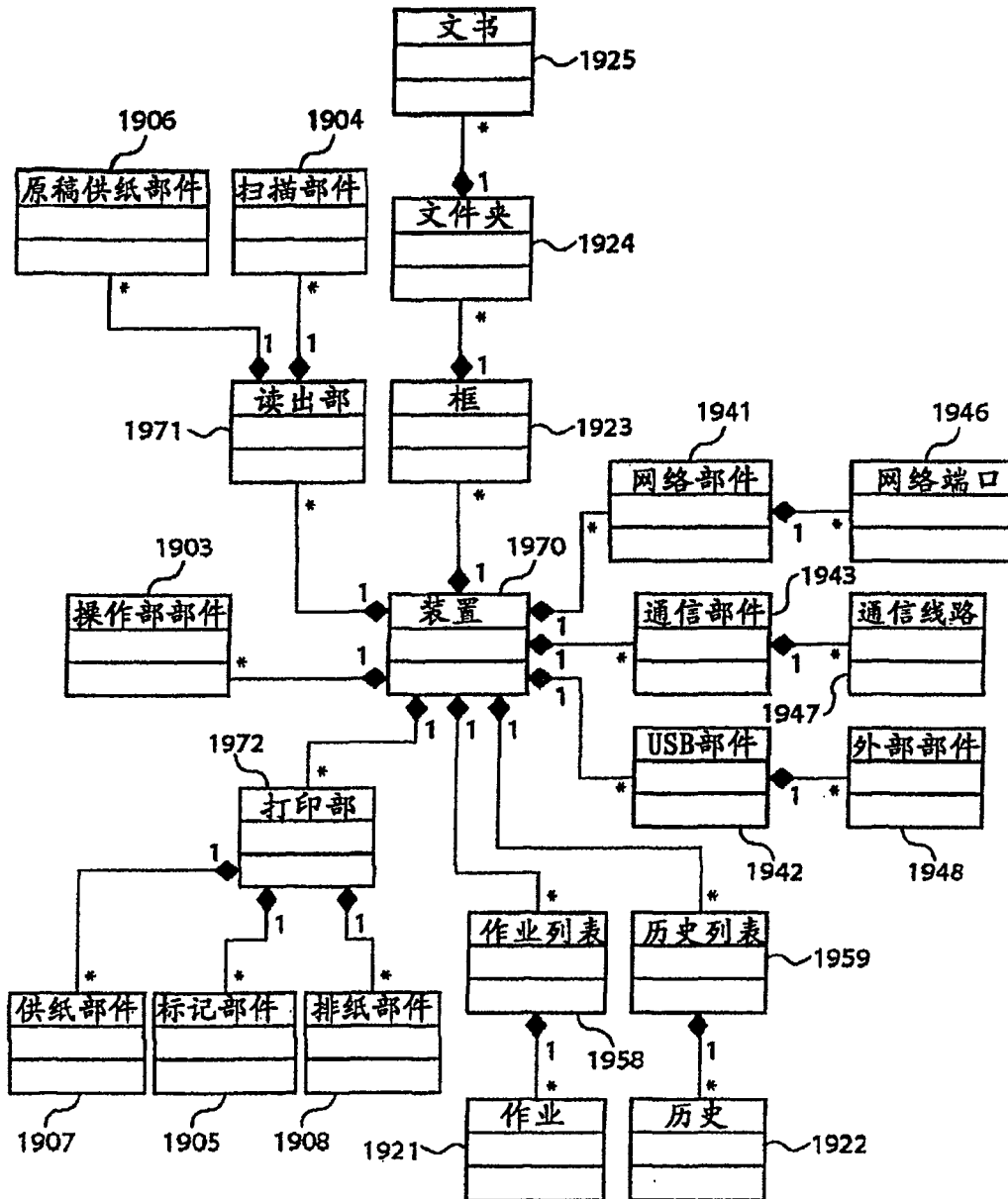


图 17

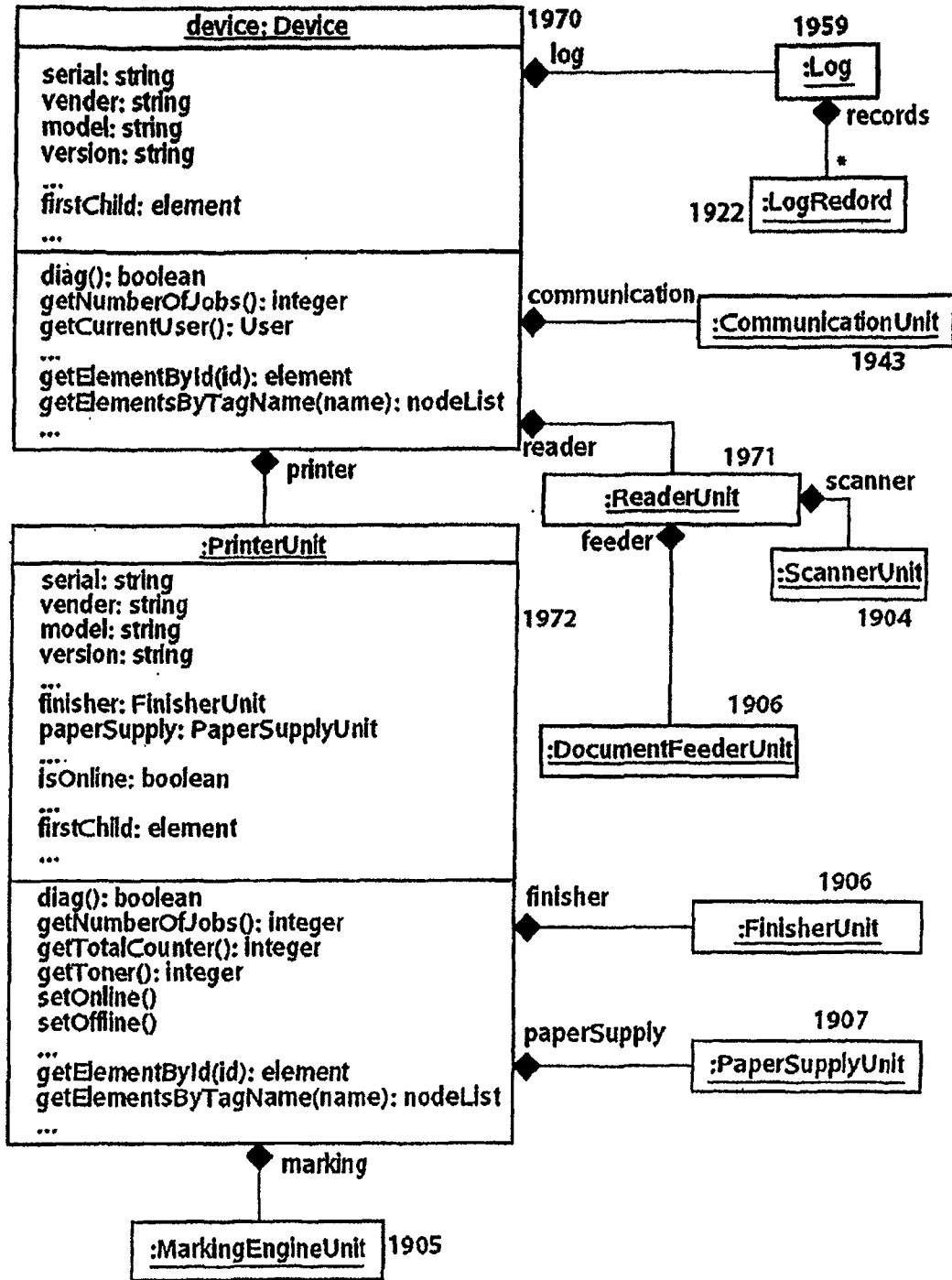


图 18

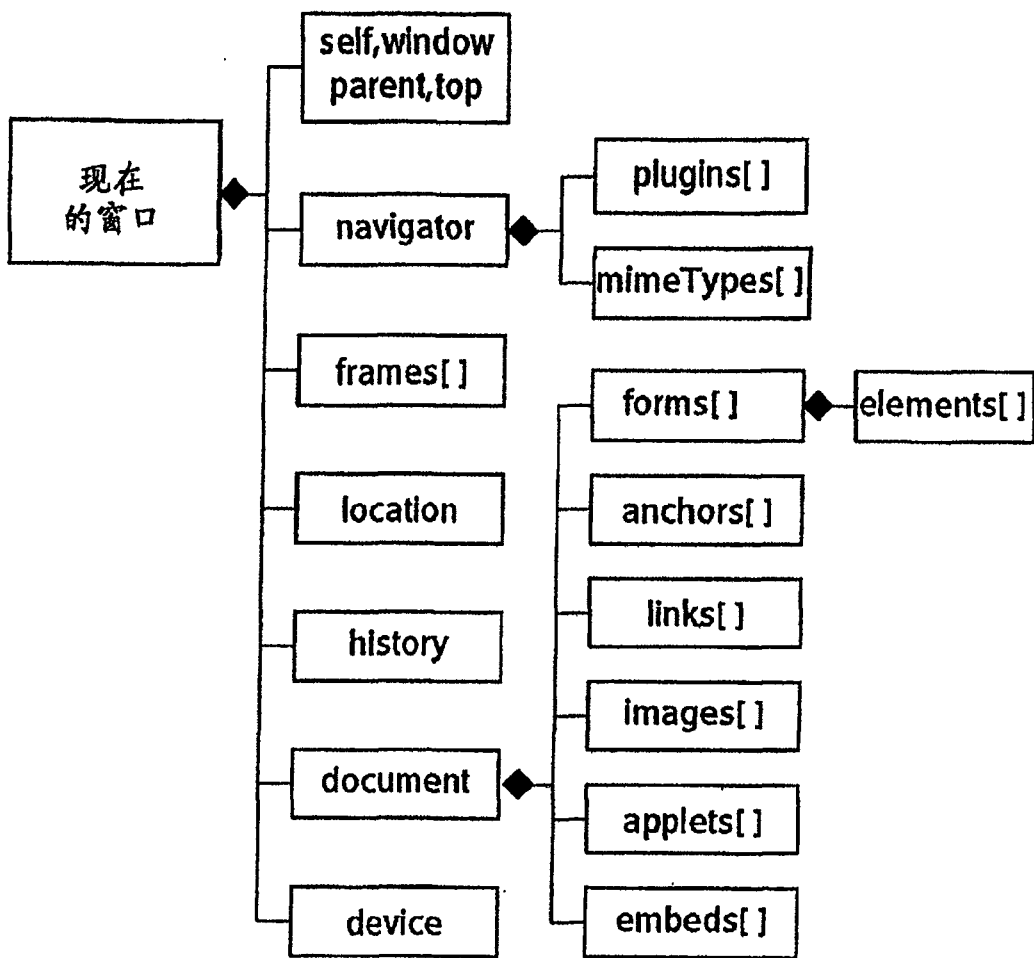


图19

```

1 <HTML>
2 <HEAD>
3 <TITLE>Customer Relationship</TITLE>
4 </HEAD>
5 <BODY>
6 <H1>
7 <SCRIPT>document.write(device.getCurrentUser().name);</SCRIPT> 欢迎
8 </H1>
9 <P> <SCRIPT>document.write(device.model);</SCRIPT> 感谢您一向的使用。
10 </P>
11 <H2> 通知 </H2>
12 <SCRIPT>
13 if (device.reader.model == "S500") {
14   if (device.reader.feeder.model == "DF03") {
15     document.writeln("<P><IMG SRC='document-insert-unit-b2.jpg' ALT='\">
16     '<IMG SRC='arrowRed.gif' ALT='\">
17     ' 这个月在期间限定中,进行原稿供纸部件向最新种类的抵价更换服务.',
18     ' 在本次机会中更换为最新模型吗? ',
19     ' 具体是<A HREF='\"http://canon.com/offer\", device.reader.model, 'feeder.html,
20     '\">, device.reader.model,
21     '用原稿供纸部件的抵价促销的通知 请阅读。
22     '<P>');
23   }
24 }
25 if (device.printer.model == "P9900") {
26   if (device.printer.finisher) {
27     document.writeln("<P><IMG SRC='arrowRed.gif' ALT='\">
28     '向您使用的打印机中追加方便的供纸部件选项吗? ',
29     '现在正在打折销售。 ',
30     '具体是<A HREF='\"http://canon.com/special-saler\", device.printer.model, 'finisher-pf30.html,
31     '\">, device.printer.model, '用排纸部件打折促销的通知 请阅读。
32     '<P>');
33 </SCRIPT>

```

图 20

```

34 <SCRIPT>
35   var latests = new Array();
36   latests["operationPanel"] = { version:"1050", label:"操作部" };
37   latests["printer"] = { version:"1011", label:"打印机" };
38   latests["reader"] = { version:"1213", label:"读出部" };
39   latests["network"] = { version:"1008", label:"网络部件" };
40   latests["communication"] = { version:"1099", label:"通信部件" };
41   latests["usb"] = { version:"1101", label:"USB 部件" };
42   for (var unit in latests) {
43     if (device[unit] && (device[unit].version < latests[unit].version)) {
44       var uri = 'mailto:support@canon.jp?' +
45         'subject=Request upgrade%20' + device.model + '%20%23' + device.serial + '&' +
46         'body=Unit%20model%3a%20' + device[unit].model +
47         '%20%28' + 'Current%20version%3a%20' + device[unit].version + '%29';
48       document.writeln('<P><IMG SRC="arrowRed.gif" ALT="">',
49         latests[unit].label, ' 能把固件更新为最新版本 。 %n',
50         '您正在使用的版本是 device[unit].version, 。 %n',
51         '最新版本, latests[unit].version, 。 %n',
52         '希望的客户 <A HREF="', uri, '"> 请用 email 申请 </A>',
53         '< %n',
54         '我们派遣客户工程师 。 <P>%n');
55     }
56   }
57 </SCRIPT>
58 <SCRIPT>
59   var toner = device.printer.getToner();
60   if (toner < 10) {
61     var uri = 'http://canon.jp/order/' + device.printer.model + '/toner.html';
62     document.writeln('<P><IMG SRC="arrowRed.gif" ALT="">注意:%n',
63       '打印机的粉末已经要用完了。(粉末剩余量: ', toner, '%) %n',
64       '一定<A HREF="', uri, '">, device.printer.model,
65       '请从纯正粉末在线订货页订货 </A> 。 %n',
66       '</P>');
67   }
68 </SCRIPT>
69 <DIV ALIGN="center">
70   <SCRIPT>
71     document.writeln('<A HREF="manual-', device.model, '.html"> 最新手册 </A>);
72   </SCRIPT>
73   <A HREF="admin.html">装置管理</A>
74   <A HREF="report.html">装置报告 </A>
75 </DIV>
76 </BODY>
77 </HTML>

```

图21

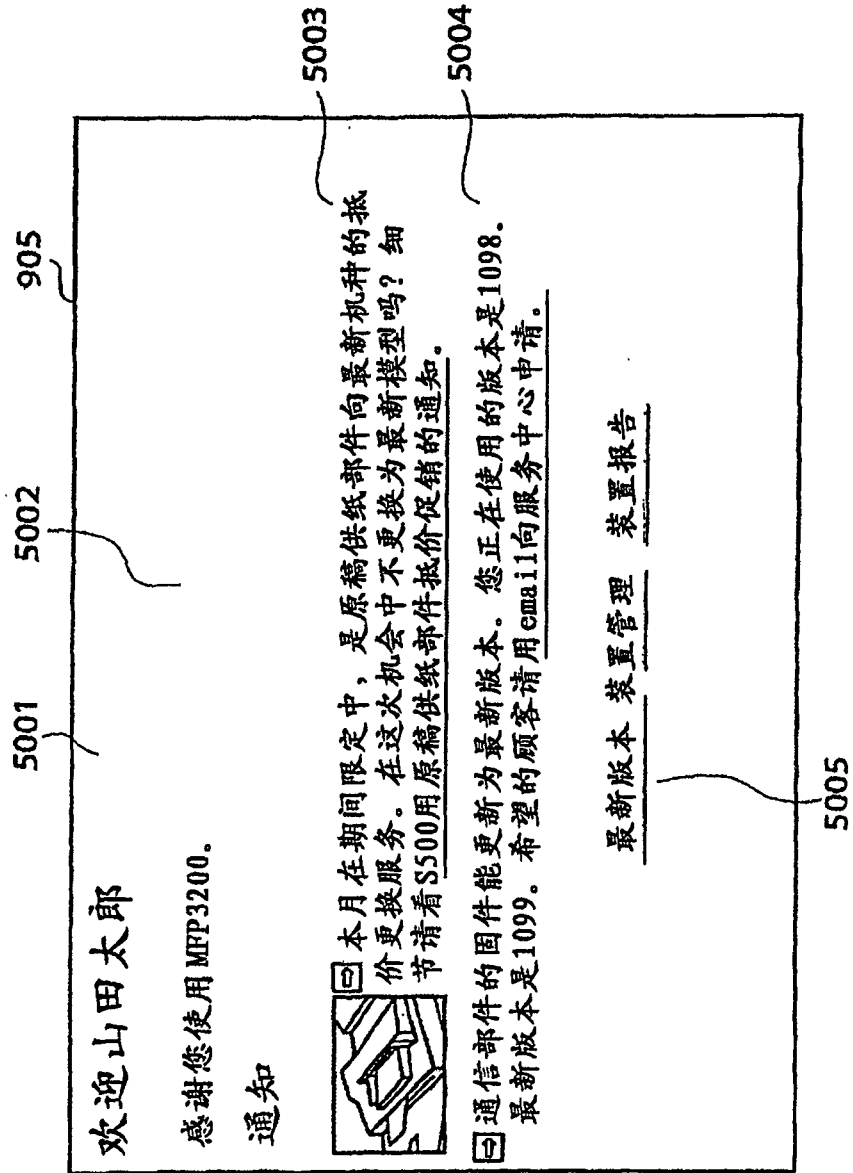



图 22


905

欢迎山田太郎


感谢您使用MFP3200。

通知

 本月在期间限定中，是原稿供纸部件向最新机种的抵价更换服务。在这次机会中不更换为最新模型吗？细节请看S500用原稿供纸部件抵价促销的通知。

 通信部件的固件能更新为最新版本。您正在使用的版本是1098。最新版本是1099。希望的顾客请用email向服务中心申请。

5101

 注意：打印机的粉末已经要用完了。(粉末剩余量:8%)
请一定从P8500用纯正粉末在线订货页订货。

最新版本 装置管理 装置报告

图 23

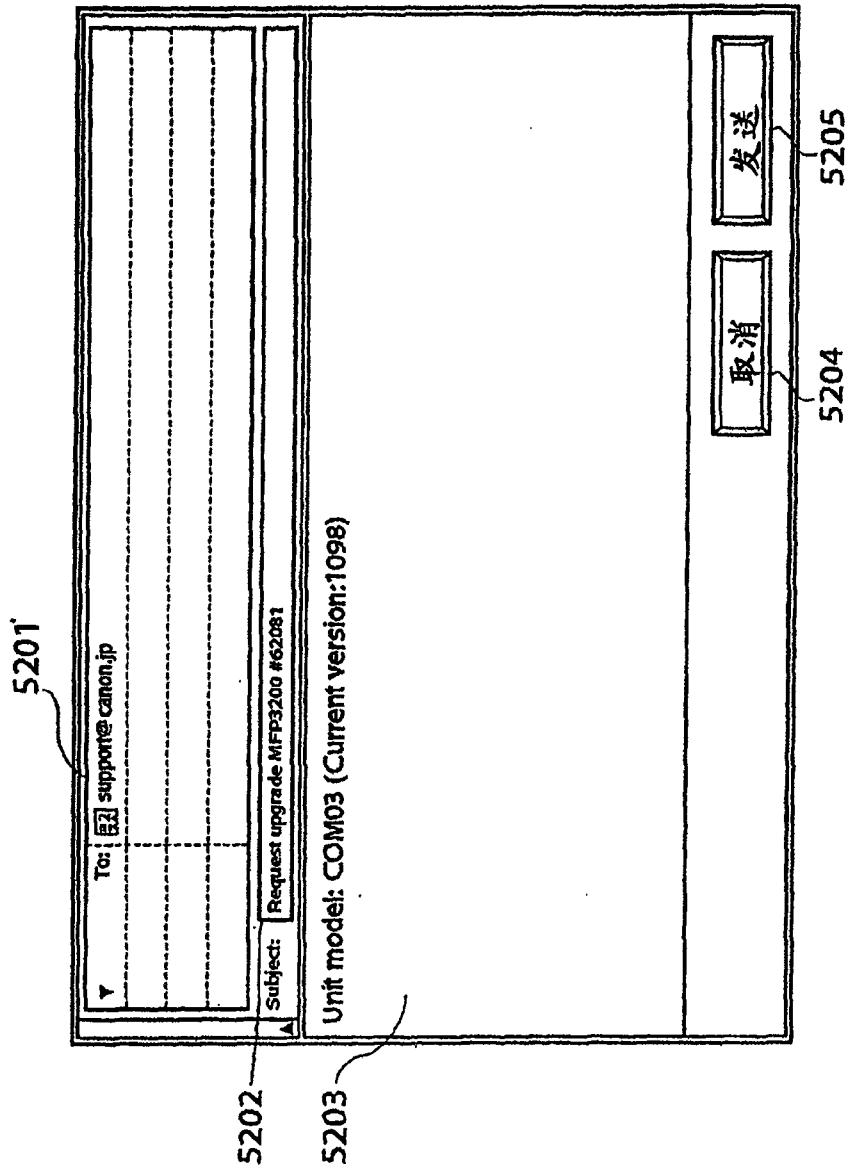


图 24

```

1 <HTML>
2 <HEAD>
3 <TITLE>MFP3200 Manual</TITLE>
4 </HEAD>
5 <BODY>
6 <H1>MFP3200 手册 </H1>
7 <P>
8 <A HREF="http://canon.jp/doc/MFP3200/knowhow.html"><IMG SRC="arrowRed.gif" ALT=""
9 BORDER="0">方便的使用技巧</A><BR>
10 <A HREF="http://canon.jp/doc/MFP3200/case.html"><IMG SRC="arrowRed.gif" ALT=""
11 BORDER="0">事例集</A>
12 </P>
13 <H2>ユニット別詳細</H2>
14 <TABLE CELLPADDING="0" CELLSPACING="0" ALIGN="right">
15 <TR>
16 <TD ROWSPAN="3">
17 <SCRIPT>
18   if (device.printer.finisher) {
19     document.write('<A HREF="http://canon.jp/manual/", device.printer.finisher.model, '.html">',
20                   '<IMG SRC="finisher-', device.printer.finisher.model,
21                   '" ALT="finisher" BORDER="0"></A>');
22   }
23   else {
24     document.write('<A HREF="http://canon.jp/order/finisher.html">',
25                   '<IMG SRC="no-finisher.jpg" ALT="Order finisher now!" BORDER="0"></A>');
26   }
27 </SCRIPT>
28 </TD>
29 <TD>
30 <SCRIPT>
31   if (device.reader.feeder) {
32     document.write('<A HREF="http://canon.jp/manual/", device.reader.feeder.model, '.html">',
33                   '<IMG SRC="feeder-', device.reader.feeder.model, '" ALT="feeder" BORDER="0"></A>');
34   }
35   else {
36     document.write('<A HREF="http://canon.jp/order/feeder.html">',
37                   '<IMG SRC="no-feeder.jpg" ALT="Order feeder now!" BORDER="0"></A>');
38   }
39 </SCRIPT>
40 </TD>

```


图 25

```

41 <TD ROWSPAN="3">
42 <SCRIPT>
43   if (device.printer.paperSupply) {
44     document.write('<A HREF="http://canon.jp/manual/,
45     device.printer.paperSupply.model, '.html">',
46     '<IMG SRC="paper-supply-', device.printer.paperSupply.model,
47     '" ALT="paper supply" BORDER="0"></A>');
48   }
49   else {
50     document.write('<A HREF="http://canon.jp/order/paper-supply.html">',
51     '<IMG SRC="no-paper-supply.jpg" ALT="Order paper supply now!" BORDER="0"></A>');
52   }
53 </SCRIPT>
54 </TD>
55 </TR>
56 <TR>
57 <TD>
58 <SCRIPT>
59   document.write('<A HREF="http://canon.jp/manual/, device.reader.model, '.html">',
60   '<IMG SRC="scanner-', device.reader.model, '" ALT="scanner" BORDER="0"></A>');
61 </SCRIPT>
62 </TD>
63 </TR>
64 <TR>
65 <TD>
66 <SCRIPT>
67   document.write('<A HREF="http://canon.jp/manual/, device.printer.model, '.html">',
68   '<IMG SRC="printer-', device.printer.model, '" ALT="printer" BORDER="0"></A>');
69 </SCRIPT>
70 </TD>
71 </TR>
72 </TABLE>
73 <P>如果触摸部件的图象，该部件的手册就打开。
74   (当未安装部件时，打开订货页)
75 </P>

```

图 26

```

76 <UL>
77 <SCRIPT>
78 document.write('<Lb><A HREF="http://canon.jp/manual/, device.reader.model, '.html">',
79 '图象读出器', device.reader.model, '</A></Lb>');
80 </SCRIPT>
81 <UL>
82 <SCRIPT>
83 if (device.reader.feeder) {
84 document.write('<Lb><A HREF="http://canon.jp/manual/, device.reader.feeder.model, '.html">',
85 '原稿供给部件', device.reader.feeder.model, '</A></Lb>');
86 }
87 </SCRIPT>
88 </UL>
89 <SCRIPT>
90 document.write('<Lb><A HREF="http://canon.jp/manual/, device.printer.model, '.html">',
91 '打印机', device.printer.model, '</A></Lb>');
92 </SCRIPT>
93 <UL>
94 <SCRIPT>
95 if (device.printer.paperSupply) {
96 document.write('<Lb><A HREF="http://canon.jp/manual/,
97 device.printer.paperSupply.model, '.html">',
98 '用纸供纸部件', device.printer.paperSupply.model, '</A></Lb>');
99 }
100 </SCRIPT>
101 <SCRIPT>
102 if (device.printer.finisher) {
103 document.write('<Lb><A HREF="http://canon.jp/manual/,
104 device.printer.finisher.model, '.html">',
105 '排纸部件', device.printer.finisher.model, '</A></Lb>');
106 }
107 </SCRIPT>
108 </UL>
109 </UL>
110 <DIV ALIGN="center">
111 <A HREF="main.html">通知 </A>
112 <A HREF="admin.html">装置管理</A>
113 <A HREF="report.html">装置报告 </A>
114 </DIV>
115 </BODY>
116 </HTML>

```

图 27

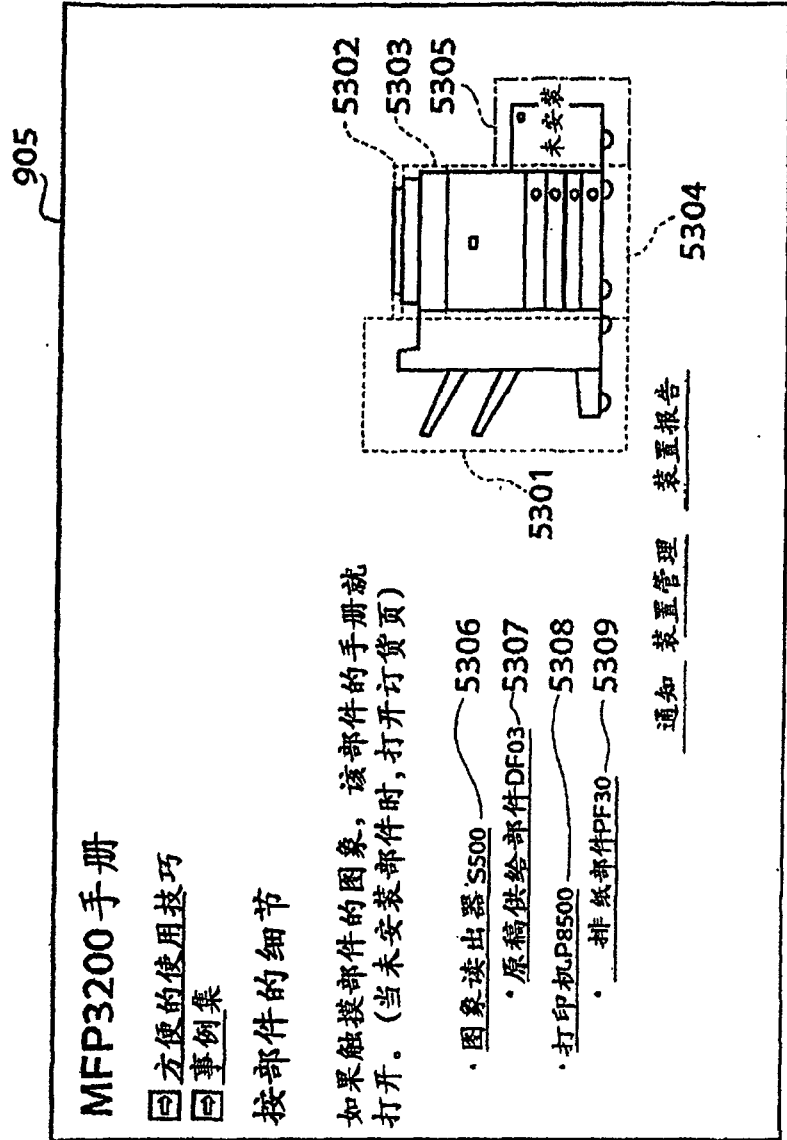


图 28

```

1  <HTML>
2  <HEAD>
3  <TITLE>Administration</TITLE>
4  <SCRIPT>
5  function diag (unit) {
6      var diagResult = unit.diag();
7      var diagDiv = document.getElementById("diagDiv");
8      var oldP = diagDiv.getElementsByTagName("P").item(0);
9      var p = document.createElement("P");
10     if (diagResult == 0) {
11         var t = document.createTextNode(' 诊断: 正常。');
12         p.appendChild(t);
13     }
14     else {
15         var uri = 'http://canon.jp/support/trouble/diag-code' +
16             diagResult + '.html';
17         var t = document.createTextNode(' 诊断: 检测到异常。 ' +
18             '错误代码是.#' + diagResult + '。更详细的信息');
19         p.appendChild(t);
20         var a = document.createElement("A");
21         var attr = a.setAttribute("HREF", uri);
22         t = document.createTextNode(uri);
23         a.appendChild(t);
24         p.appendChild(a);
25         t = document.createTextNode(' 请参照 <      。 %n');
26         p.appendChild(t);
27     }
28     diagDiv.replaceChild(p, oldP);
29 }
30 </SCRIPT>
31 <SCRIPT>
32 function showPrinterStatus() {
33     var statusDiv = document.getElementById("statusDiv");
34     var p = statusDiv.getElementsByTagName("P").item(0);
35     if (device.printer.isOnline) {
36         p.firstChild.data = ' 在线      。 %n';
37     }
38     else {
39         p.firstChild.data = ' 脱线      。 %n';
40     }
41 }
42 </SCRIPT>
43 <SCRIPT>
44 function errorHandler(event) {
45     var statusDiv = document.getElementById("statusDiv");
46     var p = statusDiv.getElementsByTagName("P").item(0);
47     p.firstChild.data = event.unit + ':' + event.message;
48 }
49 device.onerror = errorHandler;
50 </SCRIPT>
51 </HEAD>

```

图 29

```

52 <BODY onLoad='setInterval(showPrinterStatus, 10000);>
53 <H1>装置管理</H1>
54 <H2>部件诊断 </H2>
55 <DIV ID="diagDiv">
56 <P>
57 </P>
58 <SCRIPT>
59   if (device.printer.diag) {
60     document.writeln('<INPUT TYPE="button" VALUE=" 打印机" onClick="diag(device.printer);">');
61   }
62   if (device.printer.finisher.diag) {
63     document.writeln('<INPUT TYPE="button" VALUE=" 排纸部件 " ,
64       ' onClick="diag(device.printer.finisher);">');
65   }
66   if (device.printer.paperSupply.diag) {
67     document.writeln('<INPUT TYPE="button" VALUE=" 用纸供纸部件 " ,
68       ' onClick="diag(device.printer.paperSupply);">');
69   }
70   if (device.reader.diag) {
71     document.writeln('<INPUT TYPE="button" VALUE=" 读出器" onClick="diag(device.reader);">');
72   }
73   if (device.reader.feeder.diag) {
74     document.writeln('<INPUT TYPE="button" VALUE=" 原稿供纸部件 " ,
75       ' onClick="diag(device.reader.feeder);">');
76   }
77 </SCRIPT>
78 </DIV>
79 <H2>打印机状态</H2>
80 <DIV ID="statusDiv">
81 <P> 正在取得状态...
82 </P>
83 </DIV>
84 <DIV ALIGN="center">
85 <A HREF="main.html"> 通知 </A>
86 <SCRIPT>
87   document.writeln('<A HREF="manual-', device.model, '.html"> 最新手册 </A>');
88 </SCRIPT>
89 <A HREF="report.html"> 装置报告</A>
90 </DIV>
91 </BODY>
92 </HTML>

```

图 30

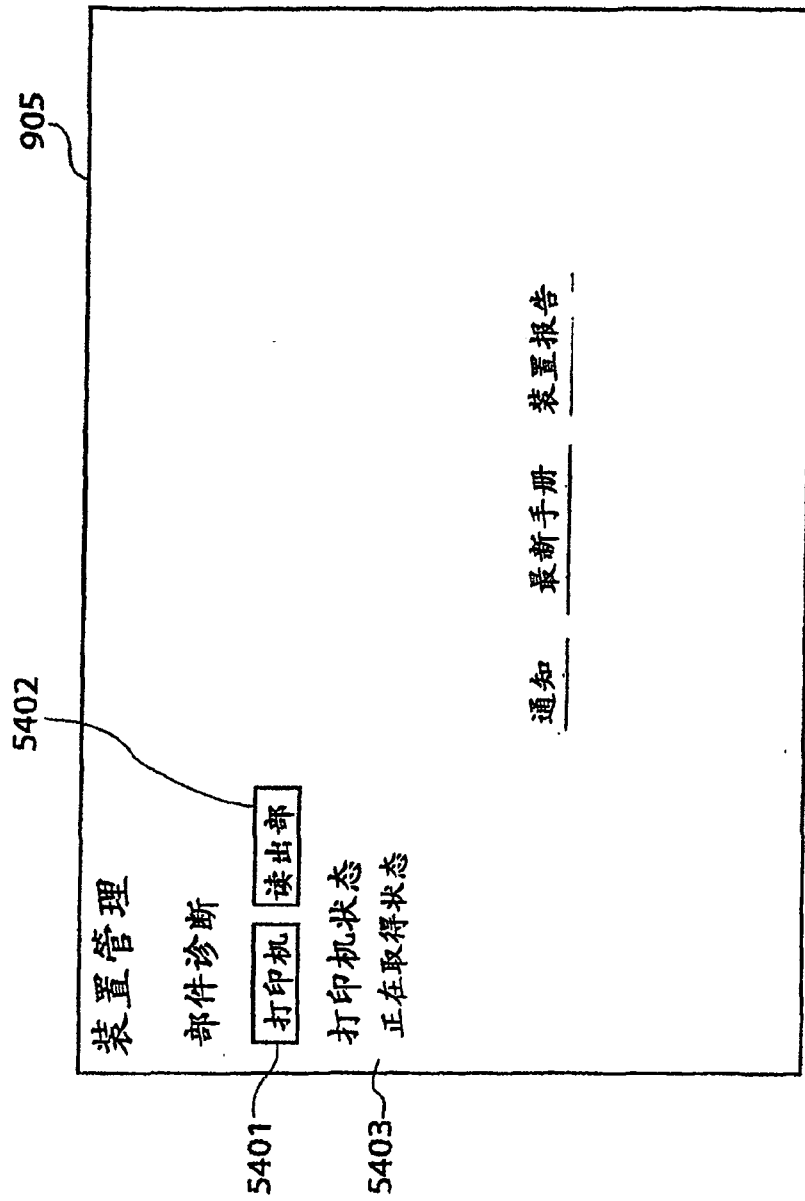


图 31

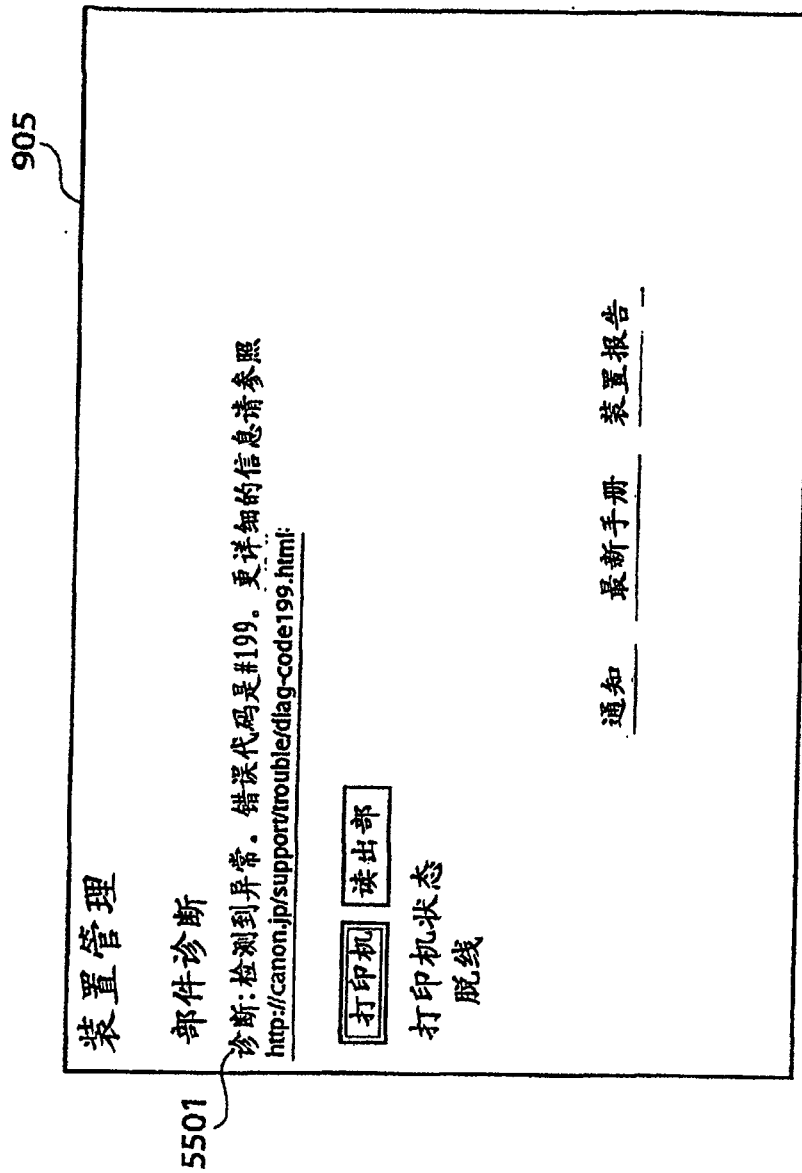


图 32

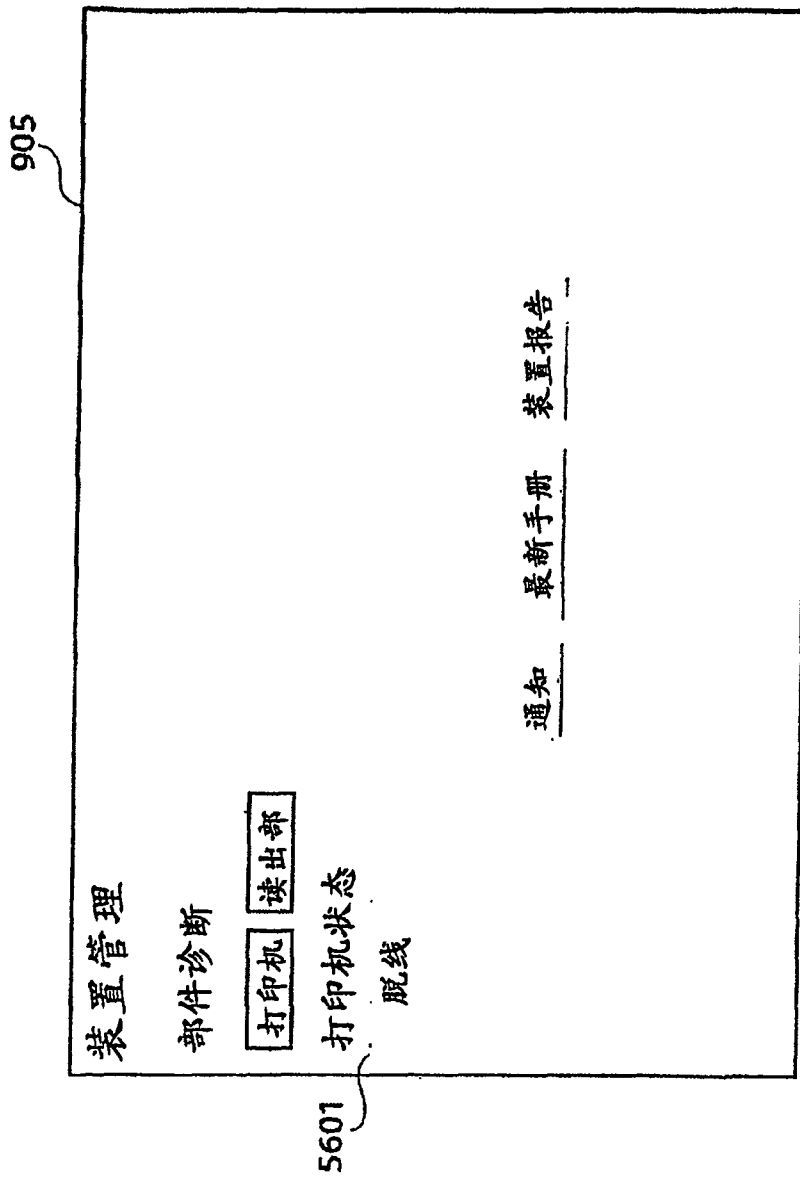


图 33

```
1 <HTML>
2 <HEAD>
3   <TITLE>Device Report</TITLE>
4   <SCRIPT>
5     function createCounterReport() {
6       document.reportForm.counter.value = device.printer.getTotalCounter();
7     }
8     function reportLog(message) {
9       document.reportForm.log.value += message + '\n';
10    }
11    function createLogReport() {
12      document.reportForm.log.value = "";
13      var log = device.getElementsByTagName("log").item(0);
14      var records = log.getElementsByTagName("record");
15      for (var i = 0; i < records.length; i++) {
16        var record = records.item(i);
17        if (record.getAttribute("facility") == "system" &&
18            record.getAttribute("level") == "notice") {
19          reportLog(record.getAttribute("date") + ': ' + record.firstChild.data);
20        }
21      }
22    }
23  </SCRIPT>
24 </HEAD>
```

图 34

```

25 <BODY onLoad="createCounterReport(); createLogReport();">
26 <H1> 装置报告 </H1>
27 <FORM NAME="reportForm" ACTION="submitReport">
28   <TABLE CELLSPACING="5">
29     <TR>
30       <TD> 计数器<BR>
31         <INPUT TYPE="text" NAME="counter">
32       </TD>
33     </TR>
34     <TR>
35       <TD> ログ<BR>
36         <TEXTAREA NAME="log" ROWS="8" COLS="70"></TEXTAREA>
37       </TD>
38     </TR>
39     <TR>
40       <TD>
41         <INPUT TYPE="button" NAME="updateButton" VALUE="更新"
42           onClick="createCounterReport(); createLogReport();">
43         <INPUT TYPE="submit" NAME="submitButton" VALUE="发送">
44       </TD>
45     </TR>
46   </TABLE>
47   <SCRIPT>
48     document.writeln('<INPUT TYPE="hidden" NAME="model" VALUE="',
49       device.model, '>');
50     document.writeln('<INPUT TYPE="hidden" NAME="serial" VALUE="',
51       device.serial, '>');
52   </SCRIPT>
53 </FORM>
54 <DIV ALIGN="center">
55   <A HREF="main.html"> 通知 </A>
56   <SCRIPT>
57     document.writeln('<A HREF="manual-', device.model, '.html"> 最新手册 </A>');
58   </SCRIPT>
59   <A HREF="admin.html"> 装置管理</A>
60 </DIV>
61 </BODY>
62 </HTML>

```

图35

905

装置报告

计数器

履历

2002/12/24	8:30 a.m.:	系统启动
2002/12/24	3:15 p.m.:	用纸滞留系统 (堵塞位置: 复制磁鼓前)
2002/12/24	3:16 p.m.:	系统状态的回归
2002/12/24	5:30 p.m.:	从无堵塞状态关闭
2002/12/25	8:30 a.m.:	系统启动

[通知](#) [最新手册](#) [装置管理](#)

图 36

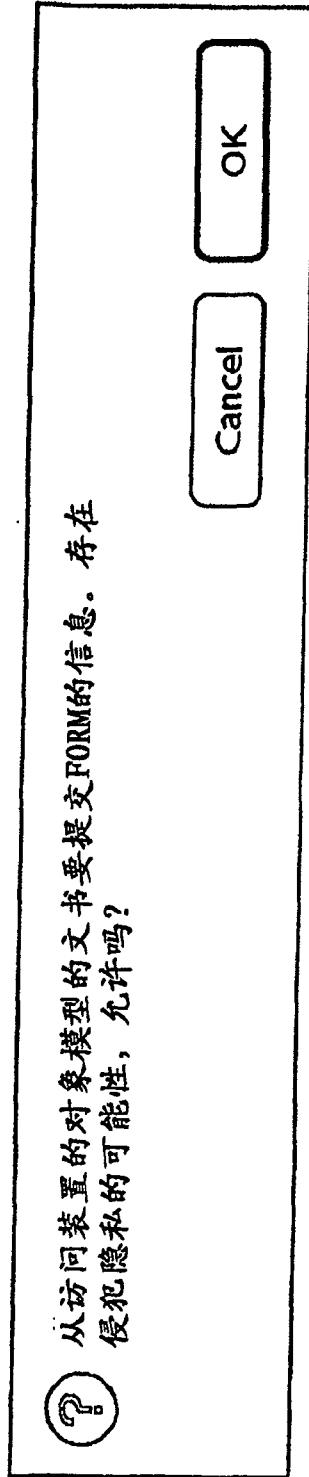


图 37

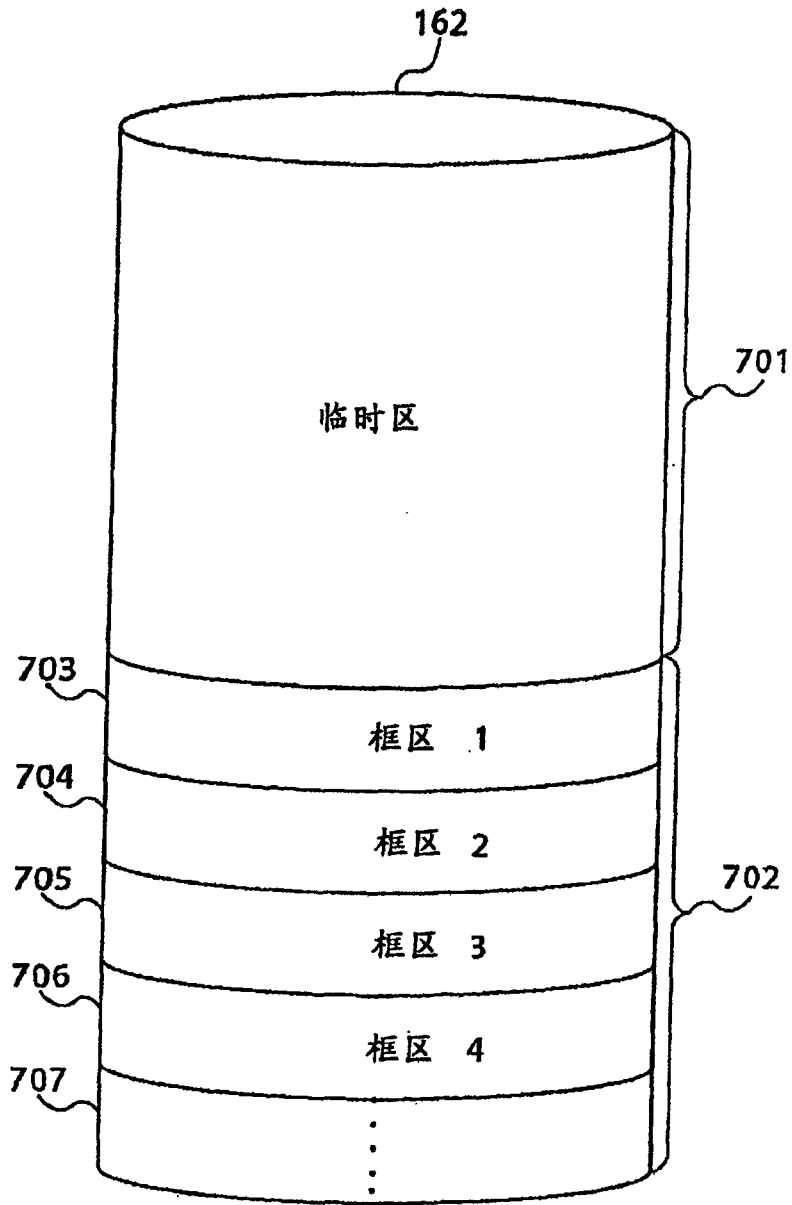


图 38

资源信息

实例

类

拥有者信息

所有者 组

访问权的设定

	读出	写入	执行
对所有者的允许	<input type="text" value="可"/> ▼	<input type="text" value="可"/> ▼	<input type="text" value="可"/> ▼
对组的允许	<input type="text" value="可"/> ▼	<input type="text" value="不可"/> ▼	<input type="text" value="不可"/> ▼
对他人的允许	<input type="text" value="不可"/> ▼	<input type="text" value="不可"/> ▼	<input type="text" value="不可"/> ▼

图 39

资源信息

实例

类

拥有者信息

所有者 组

访问权的设定

	读出	写入	执行
对所有者的允许	<input type="text" value="可"/> ▼	<input type="text" value="可"/> ▼	<input type="text" value="可"/> ▼
对组的允许	<input type="text" value="可"/> ▼	<input type="text" value="不可"/> ▼	<input type="text" value="可"/> ▼
对他人的允许	<input type="text" value="不可"/> ▼	<input type="text" value="不可"/> ▼	<input type="text" value="不可"/> ▼

图 40

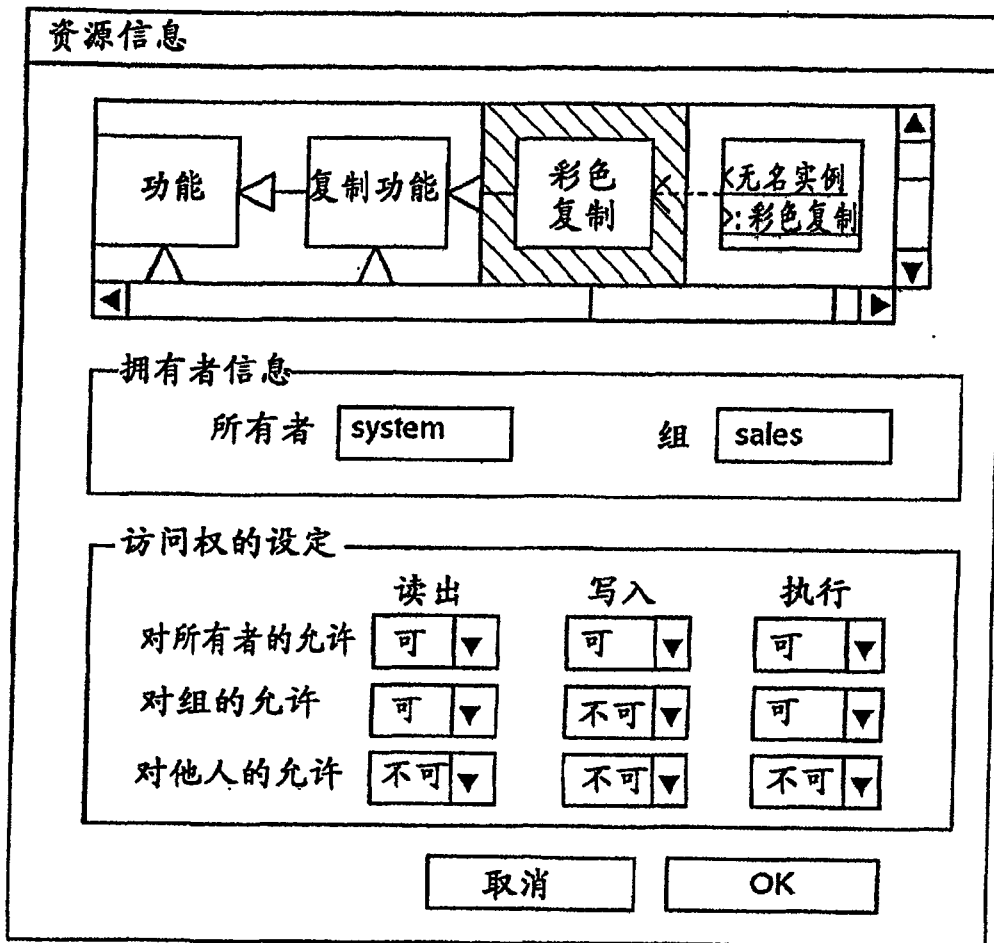


图 41

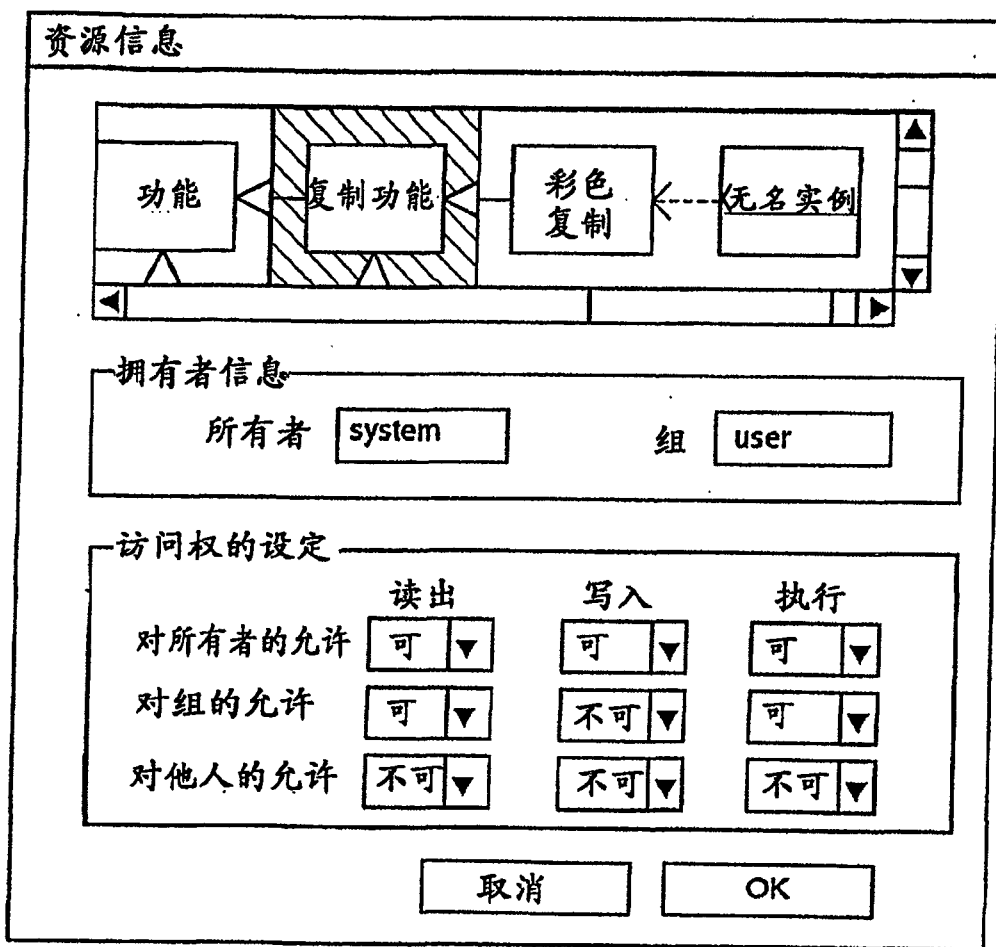


图 42

资源信息

功能 ◀ 复制功能 ◀ 彩色复制 ◀ <无名实例>:
黑白复制

所有者 system 组 user

访问权的设定

	读出	写入	执行
对所有者的允许	未设定 ▼ 可	未设定 ▼ 可	未设定 ▼ 可
对组的允许	未设定 ▼ 可	未设定 ▼ 不可	未设定 ▼ 可
对他人的允许	未设定 ▼ 不可	未设定 ▼ 不可	未设定 ▼ 不可

取消 OK

图 43

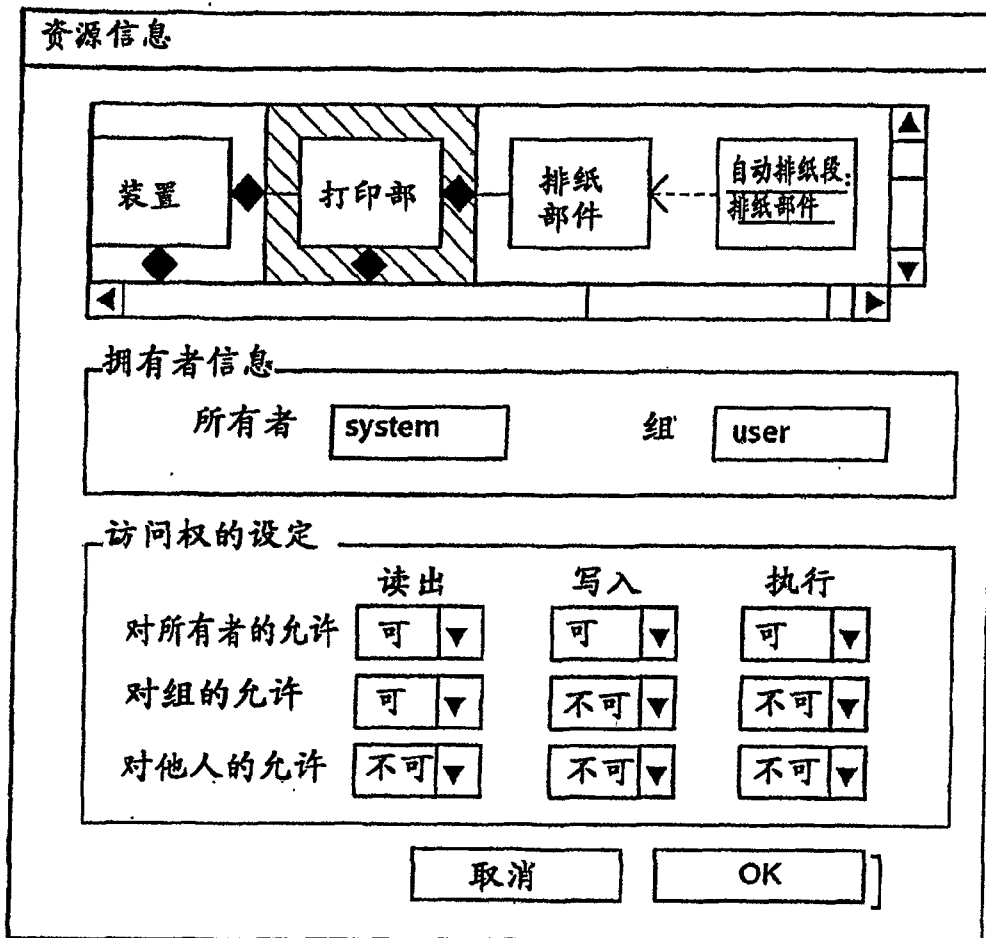


图 44

资源信息

所有者信息

所有者 组

访问权的设定

	读出	写入	执行
对所有者的允许	<input type="text" value="未设定"/> 可	<input type="text" value="未设定"/> 可	<input type="text" value="未设定"/> 可
对组的允许	<input type="text" value="未设定"/> 可	<input type="text" value="未设定"/> 不可	<input type="text" value="未设定"/> 可
对他人的允许	<input type="text" value="未设定"/> 不可	<input type="text" value="未设定"/> 不可	<input type="text" value="未设定"/> 不可

图 45

错误

因为权限不足，所以处理失败。

对于你和你所属的组，未提供执行处理所必要的对资源的访问权。

资源：技术报告书0110(类：文书)
必要的允许：读出