



[12] 发明专利说明书

[21] ZL 专利号 97111570.2

[43] 授权公告日 2003 年 5 月 14 日

[11] 授权公告号 CN 1108682C

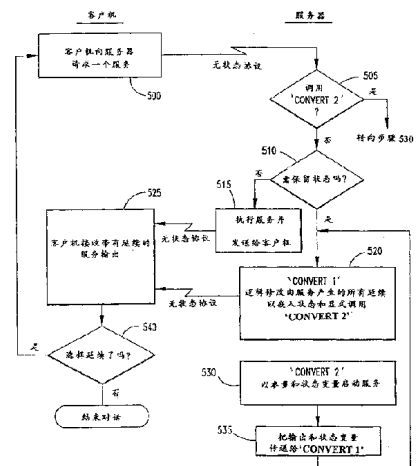
[22] 申请日 1997.5.16 [21] 申请号 97111570.2
 [30] 优先权
 [32] 1996. 6. 7 [33] US [31] 660633
 [71] 专利权人 国际商业机器公司
 地址 美国纽约
 [72] 发明人 阿伦·英加尔
 [56] 参考文献
 EP0604010A1 1994.06.29 G06F9/44
 EP0625750A2 1994.11.23 G06F9/46
 审查员 丁文勃

[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 付建军

权利要求书 4 页 说明书 24 页 附图 10 页

[54] 发明名称 在无状态网络协议情况下保留状态
 [57] 摘要

一种在使用无状态协议(例如 HTTP)的网络(例如环球网(www))上进行通信的计算机中保留状态的方法和系统。在请求服务的客户机和服务器之间的对话中保留状态是通过完成服务和识别来自服务的输出中的所有延续;并在向客户机发送的输出中的所有被识别出的延续中递归地嵌入状态信息来实现的。状态信息可以被嵌入并由服务器传送给客户机。另一种作法是,可以使用可动态下载的程序代码在客户机处嵌入状态信息。



1.在一客户机和一个或多个服务器之间的对话中保留状态信息的计算机化的方法，该客户机适于向一个或多个服务器请求服务，这些服务器通过无状态协议与客户机联成网络，所述服务包括一个或多个可能由客户机请求的数据和程序，这里的对话是客户机与一个或多个服务器之间为了所述服务而进行的通信系列，这里来自服务器的每个响应包括一个或多个延续，它们使能进行为了所述服务的另一个请求，而这里的客户机必须启动延续之一来继续对话，该方法的组成步骤是：

客户机用无状态协议发起与服务器的对话；

检测何时对服务的请求需要保留状态信息；

响应所述检测步骤，完成所述服务并从所述服务中识别出所有延续；

递归地将状态信息嵌入所有被识别出的延续；以及

响应所述嵌入步骤，将输出通信给客户机；这里在对话过程中状态信息被保留并提供给所有的服务。

2.如权利要求1所述的方法，其特征在于所述嵌入步骤由服务器完成，所述通信步骤是对所述嵌入步骤的响应。

3.如权利要求2所述的方法，其特征在于还包括一个步骤将至少一部分状态信息存储于和服务器相连的存储器中，而且所述嵌入步骤包括把代表所述状态信息部分的索引嵌入所述所有识别出的延续中。

4.如权利要求1所述的方法，其特征在于还包括一个步骤将计算机程序代码动态地下载到客户机中以完成所述嵌入步骤，它是对所述把输出通信给客户机的步骤的响应。

5.如权利要求4所述的方法，其特征在于还包括一个步骤将至少一部分状态信息存储于和服务器相连的存储器中，所述嵌入步骤包括嵌入代表所述状态信息部分的索引。

6.如权利要求1的方法，其特征在于还包括以下步骤：

客户机从所述全部识别出的带有嵌入状态信息的延续中选择出第

二个延续；以及

从所述第二延续中恢复状态信息并以恢复的状态信息启动相伴的第二服务；

递归地识别和在全部伴有所述第二服务输出的延续中嵌入状态信息。

7.如权利要求1所述的方法，其特征在于还包括一个步骤将状态信息关联于一个具体的对话。

8.如权利要求1所述的方法，其特征在于客户机和服务器是通过环球网联成网络的，其无状态协议是超文本传输协议，其延续是与超文本置标语言文件和公共网关接口程序之一的超链接。

9.如权利要求8所述的方法，其特征在于还包括一个步骤是根据预定的判据对所述超链接和从所述服务输出的数据之一进行过滤。

10.如权利要求8所述的方法，其特征在于还包括一个步骤是根据预定的判据向所述服务的所述输出增加所述超链接和数据之一。

11.如权利要求8所述的方法，其特征在于所述嵌入步骤还包括如下步骤：

修改一个识别出的延续，它是请求一个HTML文件，以所识别的延续和作为参量传送的状态信息去启动一个CGI转换器程序。

12.如权利要求8所述的方法，其特征在于所述嵌入步骤还包括如下步骤：

修改一个识别出的延续，它是一个以识别出的延续和作为参量传送的状态信息对一CGI程序的启动，这里所述嵌入步骤由该CGI程序完成。

13.如权利要求8所述的方法，其特征在于所述嵌入步骤还包括如下步骤：

修改一个识别出的延续，它是对一个CGI程序的启动，以识别出的延续、指明伴随CGI程序参量个数的参量计数器以及作为参量传送的状态信息来启动CGI转换器程序，这里所述嵌入步骤由转换器程序完成。

14.一个在客户机和一个或多个服务器之间的对话中保留状态信息

的计算机系统，该客户机适于从这些服务器请求服务，这些服务器通过无状态协议与该客户机联成网络，所述服务包括一个或多个可以由客户机请求的数据和程序，这里的对话是客户机与一个或多个服务器之间为了所述服务而进行的通信系列，这里来自服务器的每个响应包括一个或多个延续，它们使能进行为了所述服务的另一个请求，而这里的客户机必须启动延续之一来继续对话，该系统的组成是：

适于用无状态协议发起与服务器对话的客户机；

状态检测装置，用于检测何时对服务的请求需保留状态信息；

查寻装置，用于响应所述状态检测装置，从所述服务输出中识别出所有延续；

转换器装置，用于递归地将状态信息嵌入所有被识别出的延续；以及

通信装置，用于将输出传送给客户机；这里在对话过程中状态信息被保留并提供给所有的服务。

15.如权利要求 14 所述的计算机系统，其特征在于所述转换器装置由服务器执行，所述通信装置从服务器向客户机传送嵌入状态信息的输出。

16.如权利要求 15 所述的计算机系统，其特征在于还包括：

与服务器相连的存储器，用于存储至少一部分状态信息，这里所述转换器装置适于将代表所述状态信息部分的索引嵌入所述所有识别出的延续中。

17.如权利要求 14 所述的计算机系统，其特征在于所述通信装置把不嵌入状态信息的输出从服务器传送给客户机；而服务器适于把所述转换器装置动态地下载到客户机去执行。

18.如权利要求 17 所述的计算机系统，其特征在于还包括：

与客户机相连的存储器，用于存储至少是一部分状态信息；这里所述转换器装置还适于嵌入代表所述状态信息部分的索引。

19.如权利要求 14 所述的计算机系统，其特征在于客户机从所有识别出的带有嵌入状态信息的延续中选择第二个延续，该系统还包括：

转换器装置还适于从所述第二延续恢复状态信息，以恢复的状态信

息启动一个相伴的第二服务，并递归地识别和在伴有来自所述第二服务输出的所有延续中嵌入状态信息。

20.如权利要求 14 所述的计算机系统，其特征在于所述信息与具体对话相关联。

21.如权利要求 14 所述的计算机系统，其特征在于客户机和服务器是通过环球网联成网络的，其无状态协议是超文本传输协议，其延续是与超文本置标语言文件和公共网关接口程序之一的超链接。

22.如权利要求 21 所述的计算机系统，其特征在于还包括过滤器装置用于根据预定判据对所述超链接和来自所述服务的数据输出之一进行过滤。

23.如权利要求 21 所述的计算机系统，其特征在于还包括集成装置，用于根据预定的判据向来自所述服务的所述输出增加所述超链接和数据之一。

24.如权利要求 21 所述的计算机系统，其特征在于所述转换器装置还包括：

用于修改一个识别出的延续的装置，该延续是请求一个 HTML 文件，以所识别的延续和作为参量传送的状态信息去启动一个 CGI 转换器程序。

25.如权利要求 21 所述的计算机系统，其特征在于所述转换器装置还包括：

用于修改一个识别出的延续的装置，该延续是一个以识别出的延续和作为参量传送的状态信息对一个 CGI 程序的启动，其中所述嵌入操作由该 CGI 程序完成。

26.如权利要求 21 所述的计算机系统，其特征在于所述转换器装置还包括：

用于修改一个识别出的延续的装置，该延续是对一个 CGI 程序的启动，从而以识别出的延续、指明伴随 CGI 程序参量个数的参量计数器以及作为参量传送的状态信息来启动 CGI 转换器程序，其中所述嵌入操作由该转换器程序完成。

在无状态网络协议情况下保留状态

技术领域

本发明是关于计算机和计算机网络。具体地说，本发明是关于通过无状态协议（Stateless protocols）进行网络通信时由计算机保留状态。更具体地说，本发明是关于在 Internet（互联网络）上进行计算机通信时，特别是使用超文本传输协议（HTTP）通过环球网（World wide Web）进行计算机通信时，为保留状态所用的方法和系统。

背景技术

本申请题为“在无状态网络协议中保留状态”，它与下列未决（co-pending）美国专利申请相关联：

S/N 512205，1995年8月7日提出申请，题为“为分布式计算机网络建立超文本语言的方法”，申请人 Chiu 等，IBM 案卷 ST 995025 号；以及

S/N 474571，1995年6月7日提出申请，题为“通过 Internet 进行 WWW 服务器数据访问的计算机网络”，申请人 Lagarde 等，IBM 案卷 PO995020 号。这些申请具有共同的受让人，国际商用机器公司（IBM），Armonk, 纽约，这些申请在这里完全地引入作为参考。

术语表

尽管这里所用的一些术语也具有词典中的意义，但对一些术语的下列术语表可能是有用的。

Internet（互联网络）

使用一套 TCT/IP 协议的若干网络和网关组成的网络。

TCP/IP

传输控制协议/互联网络协议。是 Internet 用于分段、建立路径和重建它所处理的数据（从电子邮件到视频数据）所使用的分组交换方法。

Client（客户机）

客户机是一台计算机，它向服务器发出命令，服务器完成与此命令相关的任务。

Server (服务器)

按照另一台计算机的命令完成任务的任何计算机是一个服务器。一个环球网服务器通常支持一个或多个客户机。

World Wide Web(WWW 或 Web) (环球网)

Internet 的一种应用，它让人们通过击打感兴趣的增强亮度词或词组在 Internet 上从服务器到服务器和从数据库到数据库寻找信息（超链接）。一个 Internet WWW 服务器支持客户并提供信息。环球网可被看作是具有以 URL 编址的全部资源而且使用 HTML 显示对应于 URL 的信息并向其他 URL 提供“点击 (point-and-click)”界面的互连网络。

Universal Resource Locator(URL) (通用资源定位器)

唯一地识别和编址 Internet 信息的一种方式。可看作是电子邮件地址的环球网报告版本。如果 URL 属于埋在其他 URL 深部的报告，这些 URL 会是很累赘的。它们可由超链路访问。URL 的一例是“http://www.arun.Com:80/table.html”一个 URL 有四部分。从左边开始，第一部分指明所用的协议，用一个“:”与该定位器的其余部分分开。下一部分是目标主机的主机名或 IP 地址，它在左边用“//”分界，在右边用“/”或可选用“:”分界。端口号是可选的，左边用“:”与主机名分界，右边用“/”分界。第四部分是实际文件名或程序名。在本例中，扩展名“.html”表示这是一个 HTML 文件。

Hyperlink (超链接)

埋在你选择后被激活的词、短语、图标或图画之中的网络地址。关于那一项的信息被返回给客户机并用环球网浏览器显示。

HyperText Markup Language(HTML) (超文本置标语言)

HTML 是环球网服务器用于建立和连接由环球网客户机观看的报告所用的语言。HTML 使用超文本报告。超文本报告的其他应用在下列美国专利中描述：1993 年 4 月 20 日授予 Bernstein 等的 5,204,947 号；1994 年 3 月 22 日授予 Bernstein 等的 5,297,249 号；1994 年 10 月 11 日

授予 Lewis 等的 5, 355, 472 号; 所有这些专利的受让人均为国际商用机器公司, 这里引入作为参考。Hypertext transfer protocol(HTTP) (超文本传输协议)。

HTTP 是无状态协议的一个实例。无状态协议的意思是由客户机向服务器发出的每个请求是被独立处理的。服务器没有先前连接的记录。在 URL 的开头, “http:” 表明该文件包含超链接。

Home page (主页)

内容的多媒体表, 用于把环球网用户引向 Internet 上存储的信息, 例如关于一个组织的信息。

Web browser (环球网浏览器)

作为 Internet 旅行指南而在计算机上运行的一个程序, 当用户在 Internet “冲浪” 时使用图形桌面设备、目录和查寻工具来完成。在本申请书中 Web 浏览器是与环球网通信的客户机服务程序。

HTTP Daemon(HTTPD) (HTTP 守护程序)

具有超文本置标语言和公共网关接口能力的 IBM OS/2 环球网服务器或其他服务器。HTTPD 通常由一访问媒介 (agent) 支持, 该媒介物提供与内部网 (Internet) 上机器的硬件连接和对 Internet 的访问, 例如 TCP/IP 耦合。

Continuation (延续)

超文本链接 (或超链接) 是在客户机 - 服务器通信中 “延续” 的实例。一个 “延续” 是指一个客户机可以向服务器发出的一个新请求。每当客户机要从服务器中请求什么的时候, 服务器可以在其响应中包括一个或多个 “延续”。当一个服务器响应一个请求时, 它可以包括一个或多个 “延续”, 它们可以是任何有效的请求。然而, 有用的 “延续” 一般在逻辑上与原始请求相关联。

Conversation (对话)

客户机的服务器之间的通信序列, 在这一通信系列中服务器以一组 “延续” 响应每个请求, 而客户机总是从这组 “延续” 中选取下一个请求。在环球网上, 超文本链接代表 “若干延续”, 而一个客户机每当它

跟随超文本链接时便进入了一次对话。

网络已改变了人们使用计算机的方式。具有对一台个人计算机或工作站的访问能力的人能与 Internet 连接并与全世界的系统和人们进行通信。环球网 (WWW 或 Web) 是使用 Internet 的一种方式, 它使得用户可以通过链接的报告访问分布于全球的信息资源。WWW 还允许用户执行在远程服务器上运行的程序。这一能力使得用户能从那些由于硬件和/或软件限制而不能在当地运行的程序中获得结果。它还能下载和运行存储于远程环球网上的程序。这一点所具有的潜力是大大增加连于环球网上的计算机中的软件数量。

网络协议

网络协议提供计算机彼此通信的标准方法。协议指出数据应如何编排格式以在网络上接收和传送。异种机器能通过标准协议在网上进行无缝通信。标准的 Internet 协议的实例包括: HTTP, 参见例如“超文本传输协议 - HTTP/1.0” <http://www.ics.uci.edu/pub/ietf/http/draft-ietf-http-v10-spec-0.3.html>, 作者 T.Berners-Lee, R.Fielding 及 H.Frystyk, 1995 年 9 月 4 日; SMTP, 参见例如“简单邮件传输协议”。RFC821, J.B.Postel, 信息科学研究所 USC, 1982 年 8 月, <http://ds.internic.net/std/std10.txt>; NNTP, 参见例如“网络新闻传输协议: 一个用于以流为基础的新闻传输的建议标准”, RFC977, B.Kantor 和 P.Lapsley, 圣戈哥加州大学和伯克莱加州大学, 1986 年 2 月, <http://ds.internic.net/rfc/rfc977.txt>; FTP, 参见例如 J.Postel 和 J.K.Reynolds 的“文件传输协议 (FTP)”, RFC959, 信息科学研究所, 南加州大学, 1985 年 10 月, <http://ds.internic.net/std/std9.txt>. Gopher, 参见例如 F.Anklesoria, M.Mc Cahill, P.Lindner, D.Johnson, D.Torrey, 和 B.Alberti, “互连网络 Gopher 协议: 一种分布式报告查询和提取协议” RFC1436, 明尼苏达大学, 1993 年 3 月, <http://ds.internic.net/rfc/rfc1436.txt>; 以及 WAIS, 参见例如 F.Davis, B.Kahle, H.Morris, J.selem, T.Shen, R.Wang, J.Sui 和 M.Grinbaum, “WAIS 接口协议原型功

能说明书 (V1.5), Thinking Machines Corporation, 1990年4月。

客户机-服务器模型是网络程序设计中的主要范例 (paradigm) 之一, 参见例如 W.R.Stevens 的“Unix 网络程序设计”, Prentice Hall PTR, Englewood cliffs, Nj, 1990; 及 D.E.comer“以 TCP/IP 进行网间互联”, Vol1., Prentice Hall, Englewood cliffs, NJ. 1991, 这里全文引入作为参考。一个服务器程序提供可由多个用户通过网络进行访问的服务。当一个程序向一个服务器发出一个消息 (message) 并等待来自该服务器的响应时, 该程序便成为一个客户机。通常为用户的交互作用进行过优化的客户机处理使用所请求的服务而不必知道所请求服务或服务器的任何详细工作情况。在环球网上, “浏览器”是客户机程序, 而向浏览器回送信息的程序构成服务器程序。

客户机和服务器可以是同步通信, 也可以是异步通信。在同步通信中, 客户机在发出下一个请求之前等待来自服务器的响应。在异步通信中, 客户可在已经收到对先前向服务器的请求的一个或多个响应之前再发出请求。

许多在客户和服务器之间的网络协议是无状态的。这就是说由客户到服务器的每个请求是独立对待的。服务器没有关于先前连接的记录。HTTP 是无状态协议的一个实例。使用无状态协议的二个好处是效率高和简单。然而, 在一些场合希望客户机和服务器之间通信过程中保持状态信息。对于这些类交互作用, 协议的无状态性会出现问题。

HTTP 协议和环球网

本发明的最令人瞩目的应用是通过 HTTP 协议浏览环球网, 参见例如“超文本传输协议 - HTTP/1.0”, <http://www.ics.uci.edu/pub/ietf/http/draft-ietf-http-V10-spec-0.3.html>, 作者 T.Berners-Lee, R.Fielding, 和 H.Frystyk, 1995年9月4日, 这里全文引用作为参考。然而, 精通本门技术的人们将会理解, 本发明不限于 HTTP。现在将讨论环球网的有关方面及由协议 (例如 HTTP) 的无状态性造成的限制。

环球网由被网络联在一起的多个服务器组成。客户机通常用标准的浏览器与服务器通信, 例如由 Netscape (网景公司) 以商标“NETSCAPE

NAVIGATOR (网景导航者)”、NCSA 以商标“MOSAIC (马赛克)”或 IBM 以商标“Web EXPLORER (环球网探索者)”出售的浏览器。客户机和服务器之间通信的最普通方法是通过 HTTP 协议。HTTP 允许客户机或者通过请求一个文件或者调用一个在服务器上执行的程序 (称作公共网关接口 (CGI) 程序) 来获得数据。CGI 编程在技术上是公知的。参见例如“HTML 和发动的 (unleashed) CGI”, 作者 John December 和 Mark Ginsburg, Sams.net Publishing, Indianapolis, In.(1995)。然后服务器把文件或 CGI 程序的输出发送给客户机。服务器通常限制客户机访问文件的程序的能力。

服务器利用超文本置标语言 (HTML) 向客户机发送信息, 参见例如“HTML 原始资料集”, 作者 Ian S.Graham, John Wiley & sons, Inc., New York, 1995, 这里全文引用作为参考。HTML 报告由传统的 ASCII 文本组成, 其中要被显示的信息由 HTML 标记加以装饰。这些标记被包围在大于和小于号之间 (<... >), 告诉浏览器如何解释报告的不同部分。浏览器使用通用资源定位器 (URL) 在 Internet 上唯一地识别和寻址信息。浏览器读出对应于 URL 的 HTML 报告, 并遵循置标记中存储的指示显示这些报告。

下面给出的 HTML 代码序列 (表 1) 给出的 HTML 文本对应于 1996 年 6 月 3 日 IBM T.J.Watson Research Center 的环球网主页。这一环球网主页对应于 URL “http://www.watson.ibm.com/” 图 1 中给出一个标准的浏览器访问这一页时会显示出的对应输出。

```
<HTML><HEAD>
<TITLE>IBM T. J. Watson Research Center-home page</TITLE>
<meta name="owner" content="calyson@watson.ibm.com">
<meta name="review" content="19960202">
</HEAD>
<BODY>
<IMG SRC="/watson/mast.gif" alt="Research" >
```

```

<p>
<h1>IBM T.J. Watson Research Center</h1>
<p>
<IMG SRC="/watson/night.gif" > <IMG SRC="/watson/haw2.gif" >
<br>
<i>T.J. Watson Research Center: Yorktown (left) and Hawthorne.</i>
<p>
<ul>
<IMG align=middle SRC="/watson/bullet.gif" ><A HREF="/watwel.html" >
Welcome!</a>
<br>
<IMG align=middle SRC="/watson/bullet.gif" ><A HREF="/leo" >Local Education Outreach
</a>
<br>
<IMG align=middle SRC="/watson/bullet.gif" ><A HREF="/menu.html" > Visitor info and local
site directions </a>
<br>
<IMG align=middle SRC="/watson/bullet.gif" ><A HREF="/lodging.html" > Local hotels</a>
<br>
<IMG align=middle SRC="/watson/bullet.gif" ><A href="http://www.ibm.com"> IBM home
page</a> – <A href="http://www.research.ibm.com/"> IBM Research home page</a>
<br>
</ul>
<p>
<hr>
<A HREF="/watson/mail.html" ><IMG align=middle
SRC="/research/images/mail.gif" ></a> <b>Click on icon to send your comments.</b>
<p>
Or, contact <i>webmaster@watson.ibm.com</i>
<p>
<hr>
<Address><homepage@watson.ibm.com></address>
<b>
[
<A href="http://www.ibm.com/">IBM home page</a> |
<A href="http://www.ibm.com/Orders/">Order</a> |
<A href="http://www.austin.ibm.com/search/">Search</a> |
<A href="http://www.ibm.com/Assist/">Contact IBM</a> |
<A href="http://www.ibm.com/Finding/">Help</a> |
<A href="http://www.ibm.com/copyright.html">(C)</a> |
<A href="http://www.ibm.com/trademarks.html">(TM)</a>
]
</b>
</BODY>
</HTML>

```

表 1.对应于 IBM T.J.Watson Research Center 主页的 HTML 源代码。

许多 Web 浏览器允许用户观看被观看的任何报告的 HTML 源代码。表 1 中的 HTML 文本存储在一个文件中，该文件可被 IBM T.J.Waston 研究中心的 Web 服务器访问。当这个 Web 服务器接受一个对 URL “http://www.waston.ibm.com/” 的请求时，它向该客户机的浏览器发送适当的文件。然后该客户机浏览器将读和显示这一 HTML 文件。

（表 1 包含多个有关的链接。只有当文件存储于适当的目录中，超文本链接和图象文件才是有效的。例如，如果表 1 中的 “night.gif” 文件于任意位置，该超文本链接将会无效，于是相应的图象将不出现。

表 1 中的 “Visitor info and local site directions”（意思是“参观者所需信息及当地参观点指南”）是超文本链接（亦称超链接）的一例。图 1 中显示出会由一标准的浏览器显示出来的对应输出。当用户点击这一个由浏览器显示的链接（如图 1 所示）时，便从服务器取回一个新的 HTML 文件 “menu.html” 并由浏览器显示。与本机和远程服务器上的报告的超文本链接都能放到 HTML 文件中。能在一个 HTML 文件中包含超链接以链接全世界服务器上的报告，这种能力是环球网的主要特点之一。换句话说，利用简单的指向和点击到超文本链接上便能用 Web 浏览器访问来自全世界服务器的信息。

提醒一下，超文本链接是客户机-服务器通信中的“延续”的实例。一个“延续”是客户机可以向服务器发出的一个新请求。每当一个客户机从服务器请求什么的时候，服务器便可能在其响应中包含一个或多个“延续”。这些“延续”能代表任何有效的请求。然而，有用的“延续”通常在逻辑上与原始请求相关联。。一组好的“延续”使客户机易于与服务器进行同步通信。在每个请求之后，服务器以一组“延续”作为响应。客户机选择“延续”之一作为下一个请求。一个“对话”是客户机和服务器之间的一个通信系列，其中服务器以一组“延续”来响应每个请求，而客户机总是从这组“延续”中选择下一个请求。

在环球网上，超文本链接代表“延续”，每当客户机遵循超文本链

接时，它便进入了对话状态。每当客户机明确地请求一个新的 URL 从而得到一个页面，而不再遵循超文本链接时，对话便被中断。如果对应于被中断对话的页面仍能被客户机使用（例如存在浏览器高速缓冲存储器或磁盘存储器中），则继续被中断的对话是可能的。如果是这样，可通过重新加载该页面而继续对话并继续遵循超链接。在同一对话中客户机可以与多个服务器通信。

更正式的表述是：

如果一系列 HTML 页面 p_1, p_2, \dots, p_n 中

1. p_1, p_2, \dots, p_n 全曾被一客户机观看过，而且
2. 对所有 i ，使 $1 < i \leq n$ ，由遵循页面 p_{i-1} 上的超文本链接得到页面 p_i ，

则该 HTML 页面 p_1, p_2, \dots, p_n 系列继续对话。

在一个未被中断的对话中，客户机简单地遵循 $n-1$ 超文本链接以从页面 p_1 到 p_n ，而没有任何“反向追踪”。在一个被中断对话中，客户机至少有一次反向追踪。这里“反向追踪”的意思是：

1. 初始时观看页面 p_i ，这里 $1 \leq i < n$ ，
2. 或者遵循超链接或者明确地访问 URL 从而观看其他页面，以及
3. 通过从存储器中再加载页面 p_i （假定页面 p_i 仍是可得到的）来返回页面 p_i 。

所有对 URL 的请求都是无状态的。即使一个客户机多次请求一个页面，该服务器并不保留关于先前连接的任何历史或知识。当一客户机请求一个 HTML 文件时，客户机没有办法以该请求传送附加的信息。这样，在环球网环境下当一客户机正在浏览 HTML 文件时有在整个对话过程中保留状态信息的必要性。本发明就是针对这一需要的。

例如，考虑一个处理商务事务的服务器。为了适当地发挥其功能，服务器需要状态信息，诸如客户机的用户 ID 和对应于当前交易号的交易号。这样，当一个客户机通过遵循对话中的超链接浏览 HTML 文件时需要保留这一信息。本发明满足了这一需要。在环球网上处理状态的当前方法。

当前的一种在环球网上处理状态的方法涉及 CGI 程序的使用。客户机可通过向它传送参量来启动一个 CGI 程序。例如，命令 `http://tranman.Waston.ibm.com/cgi-bin/get-args?Var1=7&Var2=10` 传送变量 `Var1=7` 和 `Var2=10` 来启动一个 CGI 程序。期望一个客户机遵循严格的句法向 CGI 程序传送变量是一件烦人的事。更加用户友好的方法是允许用户通过 HTML “表格” 输入参量。图 2 中给出由一 Web 浏览器显示的 HTML 表格举例。用户填写适当的字段并通过点击发送钮来向服务器发送信息。由用户键入的值作为参量传送给一个 CGI 原本 (script)。客户机不需要知道被启动的 CGI 程序的详细情况或由该程序所期望的参量的格式。

表格允许客户机向服务器传送状态。服务器也能使用表格向客户机传送变量。表格中可以包含这样一些隐蔽的变量，它们不对客户机显示出来，但在客户机提交表格时回送给服务器。Web 服务器通常通过传送作为表格中隐蔽变量的状态变量来保留状态。当客户机提交表格时，接受表格的服务器能从隐蔽字段中得到的状态变量。

例如，假定一个商务事务处理服务器在与一个客户机通信。事务处理服务器需要得到客户机用户 ID 和一次会话的 ID 以便与客户机进行其余的对话。服务器能从该客户机提交的表格中得到该客户机的用户 ID。该表格启动一个 CGI 程序，然后该 CGI 程序产生一个会话 ID。其后来自服务器的每个响应都是一个表格。该表格是动态产生的，并包含用户和会话的 ID 作为隐蔽变量嵌入其中。客户机通过完成和提交由服务器产生的表格来作出响应。

图 3 描述了当前的一种使用 HTML 表格保留状态的方法。服务器 410 把隐蔽参量中的状态变量嵌入动态产生的 HTML 表格 420 中。状态变量 425 在客户机 450 和服务器 410 之间往返传送。利用表格，客户机 450 和服务器 410 往返传送状态信息 425。服务器 410 通过急速 (on the fly) 建立 HTML 表格并把状态变量 425 嵌入隐蔽字段来向客户机传送状态信息。客户机 450 通过完成和提交由服务器 410 产生的表格 420 来向服务器回送状态信息 425。

处理状态的当前技术的局限性

刚才概述的方法的问题在于它严重地限制了客户机和服务器在对话期间彼此相互作用的类型。服务器 410 必须总是以动态生成的包含隐藏变量 425 的 HTML 表格 420 来响应客户机 450。在客户机浏览 HTML 文件时没有办法保存状态。例如，假定客户机希望在会话过程中浏览一个目录。该目录由若干 HTML 文件组成。利用当前技术则没有办法既让客户机浏览目录（中的不同 HTML 文件）又不丢失状态信息。如果服务器允许客户机通过观察目录来继续对话，则当客户机一访问目录中的一个 HTML 文件便立即丢失状态信息。

这样，需要一个系统和方法，它允许客户机浏览目录，即访问不同的 HTML 文件，而又保留状态信息。本发明就是要满足这种需要，不论目录中包含的 HTML 文件是否存于不同的服务器上。

当前技术对保留状态的局限性已被其他人注意到。例如，见“持续性的客户机状态 HTTP 辅件（Cookies）”，网景通信公司，1996，http://home.netscape.com/newsref/std/cookie_spec.html；还可参见“建议的 HTTP 状态信息机制”，D.M.Kristol, AT&T 贝尔实验室，1995 年 9 月 22 日，http://www.research.att.com/~dmk/session_01.txt；以及 M.Culter 和 D.Hall，“1995 年 8 月对环球网的观察”，<http://www.netgen.com/corpinfo/press/webwtchv6ng.html>。Kristol 建立的解决办法要修改 HTTP 协议以保留状态，与此不同的是本发明保留状态而无需改变底层的协议。

由网景通信公司提出的另一个解决办法得要给它们的浏览器增加一个特征，称之为辅件；参见“持续性的客户机状态 HTTP 辅件”，网景通信公司，1996，http://home.netscape.com/newsref/std/cookie_spec.html。这里，服务器能通过在其响应的后面追加一个状态对象（称作辅件）来满足一个 HTTP 请求。这个辅件包含对该状态有效的 URL 范围的描述。这个辅件由客户机运行的 Netscape（网景）浏览器存储。由该客户机在将来向辅件中指定的 URL 之一发出任何 HTTP 请求时，这些请求中都将包含存储于辅件中的状态对象从客户机向服务器的反向传送。

这种方法有许多缺点。希望保留状态的服务器应用程序必须提供一个可能利用这一状态的所有 URL 的清单。这是烦人的事，而且有时会是不可可能的。辅件还缺少一种方法把状态信息和具体的对话关联起来。例如，假定一个浏览器在二个不同的对话中访问同一个 URL。在第一个对话过程中，在该 URL 被访问时状态信息存在并被通过辅件传送给服务器。在第二个对话过程中，在该 URL 被访问时不存在状态信息。然而，老的辅件仍然存在，而且老的状态仍然传回服务器。这会使服务器弄混淆是否相信老的状态信息仍适用于这新的对话。另一个问题是辅件不是标准特征，它将只工作于支持 Netscape（网景）协议的服务器和浏览器。

这样，需要一种方法和系统在无状态协议中保留状态，它不限于一个需要利用状态信息的 URL 清单，而且其状态信息与具体的对话相关联从而避免向服务器传送过时状态信息的问题。再有，需要一个在诸如 HTTP 之类协议中保留状态的系统，它能以任何支持 HTTP 协议的浏览器进行工作，而不需要在客户机或服务器上具有特殊的非标准的特征。

发明内容

根据前述需要，本发明是针对使用无状态协议进行网络通信的计算机中保留状态的方法和系统。尽管最佳实施例是用于使用超文本传输协议（HTTP）在环球网（WWW 或称 Web）上进行通信的计算机，但本发明也适用于其他形式的网络通信。

假定由服务器代表一客户机完成的服务是由客户机启动的程序。一个服务能接收不定数量的参量。一个对话是该客户机与一个或多个提供服务的服务器之间进行的一个通信系列，这里每个来自服务器的响应包括一个或多个“延续”，它们使能另一个对服务的请求，而且这里的客户机必须启动“延续”之一以便继续对话。

于是，一个计算机化的方法、系统和计算机程序产品，它具有本发明的特点，即在适于请求服务的客户机与一个或多个通过无状态协议与客户机联网的服务器之间的对话当中保留状态信息，它包括：客户机用无状态协议发起与服务器的对话；检测对服务的请求何时需要保留状态信息；当状态被保留时，完成服务并识别来自服务的输出中的全部“延

续”；在所有识别出的“延续”中递归地嵌入状态信息；以及将输出传送给客户机；这里对于对话过程保留状态信息并将其提供给所有的服务。

根据本发明的另一方面，状态信息的嵌入是由服务器完成的，而且由服务器将其传送给客户机。本发明的另一方面包括：将至少一部分状态信息存储于和服务器相连的存储器中，并将代表该部分状态信息的一个索引嵌入所有被识别出的“延续”中。

本发明的又一个方面包括向客户机动态下载计算机程序代码，以把状态信息嵌入到来自也与该客户机通信的服务的输出中。本发明的再一个方面包括把至少一部分状态信息存入与客户机相连的存储器中并嵌入代表所存状态信息的索引。

在一个最佳实施例中，当使用 Web 浏览器在环球网上遍历超文本链接时，我们的方法允许保留全部状态信息。超文本链接由若干个“延续”构成。客户机浏览器通过遵循超文本链接来遵循一个对话，从而取得新的页面。本发明的特点是在任何对话过程中保留状态变量。根据本发明的一个方面，在整个对话过程中被保留的状态变量被传送到整个对话过程中可能被启动的每个 CGI 程序。

当客户机和服务器是通过环球网连网时，无状态协议是超文本传协议 (HTTP)， “延续”是对超文本置标语言 (HTML) 文件和公共网关接口 (CGI) 程序之一的超链接，本发明的特点在于使能根据预定判据滤掉和/或增加来自服务的超链接和数据输出。本发明嵌入状态信息的再一个方面包括：修改一个被识别的“延续”，该“延续”是对一个 HTML 文件的请求，用以把被识别的“延续”和传送的状态信息作为参量来启动一个 CGI 转换程序。本发明嵌入状态信息的又一个方面包括修改一个被识别的“延续”，该“延续”是把被识别的“延续”和传送的状态信息作为参量对一个 CGI 程序的启动，这里由该 CGI 程序完成嵌入步骤。

本发明嵌入状态信息的另一方面包括：修改一个被识别的“延续”，该“延续”是对一个 CGI 程序的启动，以用识别出的“延续”、一个指示伴随该 CGI 程序的参量个数的参量计数器以及所传送的状态信息作为参量来启动一个 CGI 转换程序，这里的嵌入步骤由该转换程序完成。

附图说明

下面结合附图进行的描述将使本发明的这些或那些特点和优点更加显然，这里，

图 1 是由标准浏览器显示的 HTML 页面一例；

图 2 显示由 Web 浏览器观看到的 HTML “表格” 的一例。

图 3 显示出用 “表格” 保留状态变量的客户机和服务器的方框图；

图 4 是根据本发明使用无状态协议在网上保留状态的一般方法的方框图；

图 5 是通过无状态协议传送的数据包的一般化框图；

图 6 是根据本发明保留状态信息而对图 5 的数据包进行修改后的实例；

图 7a 描述具有本发明特点的一个系统的实施例，用于在环球网上传送商务业务并保留状态；

图 7b 描述在图 7a 系统上保留状态的方法的实施例；

图 8 描述根据本发明的方法的更详细举例，用于在图 7a 所示系统中和在图 7b 的方法步骤 745 中保留状态；

图 9a 描述连于 HTML 文件的一个超文本链接的结构；

图 9a' 描述图 9a 所示结构，但根据本发明进行修正以保留状态；

图 9b 描述与 I 类 CGI 程序的超文本链接的结构；

图 9b' 描述图 9b 所示的结构，其中根据本发明嵌入了状态参量；

图 9c 描述与 2 类 CGI 程序的超文本链接的结构；以及

图 9c' 描述图 9c 所示的结构，但根据本发明进行了修改以保留状态。

具体实施方式

在使用无状态协议的对话中保留状态的方法的详细描述

本发明是使通过无状态协议在网络上通信的计算机具有能力的技术。虽然最佳实施例是针对使用超文本传输协议 (HTTP) 在环球网 (WWW 或称 Web) 上通信的计算机，但本发明也适用于其他形式的网络通信。

图 4 描述了根据本发明用无状态协议保留状态的一般方法，即它不

限于环球网上的客户机和服务器通信。下文中要描述的最佳实施例具体应用于环球网，这里为了简化而假定由服务器代表一客户机完成的服务是由客户机启动的程序，并假定一个服务能接受可变个数参数。

如图所示，在步骤 500，一客户机请求来自服务器的一个服务。当所提供的服务不需要保留状态信息时所采取的路径由步骤 505、510 和 515 代表。在步骤 510，在某一点上服务器处理一个请求，对这一请求服务器确定要使当前对话中会被启动的全部服务能得到所需要的状态变量。于是服务器将其输出连同所有的状态变量（由<state-variable-list>（状态变量表）表示）传送给由 convert1 表示的程序，并使过程继续到步骤 520。在步骤 520，convert1 程序修改由该服务产生的“延续”，但把其他数据不做修改地传回给客户机。

例如，如图 5 中所示，在正常环境下，代表对程序 service1 的调用的一个“延续”会有如下形式：

service1<service-arg-list> (1)

这里 service1605 是一个服务，<service-arg-list> 是如果该客户机选择该“延续”则传送给该服务的服务参量表 610。如图 6b 所示，本发明的 convert1 程序通过把每个“延续”（1）修改成如下形式来保留状态：

convert2 service-string<state-variable-list> (2)

这里 convert2 650 是对一项具体服务的调用（下文中将描述），service-string 680 是包含 service1 605、<service-arg-list>610 和一些定界信息 685 的串，以把服务参量 610 与图 6 中所示对 convert2 调用中的<state-variable-list>670 区分开。<state-variable-list>670 代表在该对话期间要保留的且可被所有服务使用的状态信息。

再参考图 4，在步骤 525，客户机接受由服务器发出的输出和修改过的“延续”。现在，该对话的每个修改过的“延续”是如（2）式所示的对 convert2 程序的调用。在步骤 540，客户机检验该输出。如果一个“延续”被选择到，则过程返回步骤 500，在那里该（修正过的）服务请求被送到服务器。在步骤 505，服务器处理修正过的服务请求并启动 convert2 程序，并在步骤 530 继续处理过程（由于在前一步骤 520 中明

确嵌入的调用)。在步骤 530 (参考图 5 和图 6), convert2 程序对服务 605 和将被由 service-string 680 传给服务器的参量 <service-arg-list>610 进行句法分析。convert2 程序 605 通过把 <service-arg-list>610 以及 <state-variable-list> 上的所有变量传送给所请求的服务 605 (这里是 service1) 来启动服务 605。这样, service1 服务在需要时能访问所有的状态变量。在步骤 535, convert2 程序接收服务 (service1) 输出并把该输出连同 <state-variable-list> (状态变量表) 传送给 convert1。在步骤 520, convert1 修改每个“延续”(如前文所讨论的那样)。在步骤 525 再次将输出传送给客户机并重复其处理过程, 使在对话过程期间保留状态信息 670。

图 7a 描述具有本发明特点的一个系统的实施例, 用于在环球网上传送商务业务。Web 服务器 410' 允许在 Internet (互联网络) 上进行售货商务活动。消费者通过运行标准浏览器 460 的客户机 450 来访问服务器 410'。为了安全地通信, “浏览器” 460 应能用 SSL 通信。例如参见 A.O.Freier, P.Karlton, P.c.Kocher 的“SSL 协议 3.0 版”, Internet 草案, 1996 年 3 月, <http://home.netscape.com/eng/ssl 3/ssl-toc.html>, 这里全文引入作为参考。然而, 不支持 SSL 的浏览器 10 也能使用一些服务。用可以浏览目录, 这些目录可以存储在一个稳定的存储介质上, 例如直接存取存储装置 (DASD) 470。象使用传统目录那样, 用户浏览产品说明并能选取中哪些项目加到他们的购买目录中。当用户已确定完成购买目录, 他将进行购买并接着付账。

如图中所示, 服务器 410' 可以包括一个传统的数据库管理系统 (DBMS) 412 来管理数据库 475 中所存储的关于, 消费者、存货、和产品的信息。DBMS 的一个实例是 IBM 公司以商标“DB2”销售的 DBMS。此外, 服务器 410' 允许用户在对话过程中浏览产品目录。服务器 410' 对产品目录的格式限定很少。目录可由 HTML 文件 425 以及传统的 CGI 程序构成。这些文件和/或程序可与本地的或远程的服务器相关联。在对话期间必须在服务器 410' 和客户机 450 之间保持状态信息, 例如用户 ID 和会话 ID。本发明提供一种改进的方法和系统以在对话过程中透明地保

持这一状态信息。

每个客户机 450 可以访问伴随服务器 410' 的“主页 (home page)”以及观看产品目录。为了购买产品、更新消费者信息、或访问某些类型信息，用户必须通过输入用户 ID 和口令来提供证明。根据本发明，每次对话只需证明一次。一旦用户已被确认，则由本发明的转换器 416 将用户 ID 嵌入 (保留) 到对话中。

图 7b 描述了根据本发明的一种方法的实例，用于客户机 450 使用 HTTP 与服务器 410' 交互作用，同时保留状态。如图中所示，在步骤 700，客户机访问驻留在服务器 410' 上的主页。在步骤 710，客户机开始浏览产品目录，在步骤 720 继续浏览目录所提供的内容，例如从在线产品目录中选择超链接。由于只浏览目录不需要确认用者，因而通信是无状态的，而且使最大量的人们可以浏览目录。在步骤 730，找到了一项产品要加到购买清单中。在步骤 740，于是客户机必须输入用户 ID 和口令才能继续。如果该客户机对系统而言是新的，客户机选取一个用户 ID 和口令，并向服务器提供一些附加的可选信息 (地址、电话号码等)。在步骤 745，根据本发明，转换器 416 把用户 ID 和会话 ID 嵌入对话中。在步骤 750，用者能观看增加的产品、把增加的项目加到购买清单中、认可购买、或观看和更新数据库信息。保留状态变量是有好处的，从而不需再次确认用者。状态信息 (即用户 ID 和会话 ID) 将被保留，并且，在该对话的其余部分中被启动的每个 CGI 程序都能使用这些状态信息。

提醒一下，当使用当前的“表格”技术时，用者得要在每次需要确认用者的动作 (例如试图向购买清单中增加一个新项目) 之时再次输入用者 ID 和口令。会话 ID 甚至会带来更大的困难，因为服务器得让客户机记住会话 ID 并在需要确认用者时输入它。

还要提醒的是，使用“辅件”限制得保留状态的 URL 的范围。使用辅件更加缺乏把状态信息与具体对话相关联的能力，而这种能力可能使更新的状态信息提供给服务器。最后，辅件要求使用特定的浏览器，还可能要求在其客户机或服务器上具有特殊的和/或非标准的特点。

图 8 描述根据本发明的方法的更详细的举例，用于在图 7a 所示系

统中及在图 7b 的方法步骤 745 中保留状态。假定服务器 410' 是一个传统的 Web 服务器，包括典型的 Internet 联接的访问，例如 TCP/IP 耦合，进一步还有 HTML 和公共网关接口（CGI 能力 413）。

如图中所示，在步骤 740'，假定一客户机 450 运行 Web 浏览器 460，选择一起链接通过（无状态协议）HTTP 向 Web 服务器 410' 请求服务。在步骤 810，服务器 410' 把 URL 解释为例如调用一个 CGI 程序“P1” 415，它决定应在对话中嵌入状态变量，例如“ x_1, x_2, \dots, x_n ”，从而使对话中会被启动的所有 CGI 程序能给予访问。P1 为客户机 450 产生一个 HTML 页面“h”，以继续对话。不是无修改地把输出页“h”返回给客户机，而是更新“P1”以便通过把参量“ h, x_1, x_2, \dots, x_n ”传送给转换器 416 的 convert1 模块来启动本发明的转换器程序 416。对 convert1 的调用可以是如下形式：

`convert1'h,x1,x2,...,xn'`

在步骤 811 和 812，转换器程序的 convert1 模块修改在 h 中所有与 HTML 的超文本链接以保留状态变量。所有的相对超文本链接都转换成绝对超文本链接（也称为超链接）。参见前面提到的和作为参考引入的未决美国专利申请序列号 512205，1995 年 8 月 7 日提出申请，题为“为分布式计算机网络建立超文本语言的方法”，作者 Chiu 等，IBM 案卷 ST995025 号，作为从相对地址到绝对地址转换方法的一例。如所说明的那样，这些申请有同一受让人，国际商用机器公司，Armonk，纽约。那些精通本门技术的人们将会承认，能在本发明的范围和精神内以单向传送或双向传送过程来完成修改 HTML 文件的链接（步骤 811）及修改 CGI 程序的链接（步骤 812）。

如图中所示，在步骤 811 中转换器程序 416 的 convert1 模块取 HTML 页面 h 并修改所有与 HTML 文件的超文本链接以保留状态变量。与 HTML 文件的超文本链接可修改成以参量 h, x_1, \dots, x_n 调用 CGI 程序 convert2 参考图 9a，例如考虑包含下列对 HTML 文件 915 “mail.html”的参考，并假定状态变量 930 为 $x=32$ 和 $y=45$ ：

将被 convert1 逻辑修改成图 9a'描述的形式:

<A HREF=

“http://www.watson.ibm.com/cgi-bin/convert2?url=
//www.watson.ibm.com/mail.html&x=32&y=45” >

在步骤 812, 转换器程序的 convert1 模块修改所有与 CGI 程序的超文本链接。请注意, 调用 CGI 程序的超文本链接可以有两种方式保留的状态变量:

(a) 如图 9b'中所示, 把状态变量 930'传送给 CGI 程序 940, 但不把状态变量嵌入任何由 CGI 程序产生的超文本链接, 即不嵌入一个对 convert2 的调用。利用这一作法, CGI 程序负责在它产生的超文本链接之内传播状态信息; 或者

(b) 最好是如图 9c'中所示, 把状态变量 930'传送给 CGI 程序 950 并 (通过嵌入的对转换器 920'的调用) 把状态变量嵌入由该 CGI 程序产生的超文本链接之中。

为了利用二种作法 (a) 和 (b) 的好处, 转换器可以确定如何区分基于本发明范围内的多种技术中任何一个的 CGI 程序。例如, 如图 9b 中所示, 可以使用一种命名约定, 这样, 任何以子串 “type” 作为名字开头的 CGI 程序可以认为是类型 I 的 CGI 程序并用第一种方法 (a) 处理。任何不以子串 “type” 作为名字开头的子串可以被认为是类型 II 的 CGI 程序并用第二种方法 (b) 处理。

例如, 考虑类型 I 的 CGI 调用的下列举例 (参考图 9b):

<A HREF= “http://www.watson.ibm.com/cgi-bin/type1?
arg1=55” >

假定状态变量是 x=32 和 y=45。转换器 416 会把状态变量 930'附加到超文本链路, 构成如下形式 (如图 9b'所示);

<A HREF= “http://www.watson.ibm.com/cgi-bin/type1?
&arg1=55&x=32&y=45” >

现在参考图 9c 来考虑一个与类型 2CGI 程序超文本链接的例子：

```
<A HREF= "http://www.watson.ibm.com/cgi-bin/prog?
arg1=55" >
```

仍假定状态变量 930”是 $x=32$ 和 $y=45$ 。转换器会修改这一超文本链接为如下形式（如图 9c’所示）：

```
<A                                     HREF=
“      http://www.watson.ibm.com/cgi-bin/convert2?
url=//www.watson.ibm.com/cgi-bin/prog&numargs =1&arg1
=55 & x= 32&y=45” >
```

这里的参量 970“numargs=1”向 convert2 指明 CGI 程序 950 当初只有一个参量传送给它，而其余的参量是由转换器传送的状态变量 930”。然后这修改过的输出被返回给发请求的客户机。

在步骤 813，客户机 450 从服务器 410’接受 HTML 文件 h。现在，返回到客户机的每个超文本链接（唯由类型 1CGI 程序造成的超文本链接除外）都是对转换器的 convert2 例程的调用。在步骤 814，运行浏览器 460 的客户机 450 选择超文本链接之一。在步骤 815，服务器确定所选择的超文本链接是否为对 convert2 的调用。如果是，则过程在步骤 817 继续。在步骤 817，有二种可能性。

（1）传送给 convert2 的 URL 参考一个 HTML 文件。这里，过程在步骤 819 继续。例如假设客户机已选择了如下链接：

```
<A                                     HREF=
“      http://www.watson.ibm.com/cgi-bin/convert2?
url=//www.watson.ibm.com/mail.html&x=32&y=45” >
```

在步骤 819，convert2 取出文件“mai.html”中包含的 HTML 页面。然后它把 HTML 页面及状态参量 $x=32$ 和 $y=45$ 传送给转换器的 convert1 模块，然后如前文所述返回步骤 811。

或者，

（2）该超文本链接是对一个 CGI 程序的调用。在这种情况下，

过程继续到步骤 818。例如假定客户机已选择了如下链接：

```
<A HREF=
“ http://www.watson.ibm.com/cgi-bin/convert2?
url=//www.watson.ibm.com/cgi-bin/prog&numargs =1&arg1
=55 &x=32&y=45” >
```

这里，给 `conrert2` 的第二个参量 `numargs=1` 指明当初的超文本链接只向“`prog`”传送一个参量，即“`arg1=55`”。另外两个参量“`x=32`”和“`y=45`”是由转换器 416 嵌入的状态变量。`convert2` 把全部三个参量传送给 `prog`，其中包括状态变量。然后如前所述过程返回步骤 811。

本发明的这一方法通过在客户机 450 和服务器 410 之间往返传送的所有超链接中嵌入状态有利地保留了状态信息。那些精通本门技术的人们将会承认，由于把大部分状态信息存于一个和服务器 410' 相连的文件系统或数据库 425，从而可以降低客户机和服务器之间通信的详细程度。在这种情况下，只需要在客户机和服务器之间往返传送对状态变量的索引（或指针）。

本发明被设计成对标准浏览器 460 工作，它不需支持从服务器下载其后能在客户机上执行的程序。对于一个支持可下载的服务程序（例如用 Java 写成的程序（“应用部件”（applets）），或用任何其他这类语言写成的程序）的浏览器，附加特点是可能的。Java 编程环境是众所周知的技术。例如可参见 P. Tyma, G. Torok 及 T. Downing 的“Java 入门及其他 (Java primer Plus)” Waite Group Press, 1966, 这里全文引入作为参考。还可参见 Patrick Naughton 的“Java 手册”，Osborne Mcgraw-Hill, 1996, 这里全文引入作为参考。例如，服务器 410' 可以包含一个可下载程序，它使状态存储于客户机。利用这一作法，全部或部分状态能储存在客户机上。指示存储器中状态信息位置的索引（如前文说明的）可在服务器和客户机之间往返传送，以允许从客户机中提取状态。

可下载服务器代码对本发明的另一应用能允许“转换器” 416

在客户机上运行。这里，客户机将要从服务器 410' 向该客户机下载全部或部分“转换器”逻辑 416 以便执行。这将允许以全部（或部分）处理过程在本地的客户机 450 上进行来完全地实现本发明的功能。在对话过程中客户机不再不得已地走通远程服务器去过滤 HTML 页面；所有过滤可发生在本地。其优点是减小了服务器上的负载。此外，即使客户机从中取得应用部件的服务器停机或由于网络故障而不能应用，客户机也将能继续对话。

在多通信服务器上保留状态

精通本门技术的人们将会承认，在本发明的范围内可以使用多个转换器在多个服务器上传播状态。例如，在环球网上的航空预订系统可能有一个转换器（转换器 A）保留状态。超文本链路之一可能是带有自己的转换器的远程服务器上的一个旅馆预订系统（转换器 H）。一个客户机可能开始使用航空预订系统。在某点上状态信息被加到对话上。然后该客户机遵循一超文本链路达到旅馆预订系统。当客户机使用旅馆预订系统时转换器 A 继续保留状态信息。所有状态变量被传播给旅馆预定系统的 CGI 程序。这些远程服务器 CGI 程序可能简单地忽略这些状态变量。另一方面，如果旅馆预订系统理解来自航空预订系统的状态变量，这些变量有被旅馆预订系统（转换器 H）使用。

在某一点上，旅馆预订系统服务器可能启动其转换器（转换器 H）以嵌入附加状态变量。当发生这种情况时，对转换器 H 的调用可以被嵌入对转换器 A 的调用之中。这将不存在问题。CGI 程序现在将从转换器 A 和转换器 H 二者传送参量。如果转换器 A 有能力理解代表对转换器 H 的调用的 CGI 函数，额外的事情是可能的：

（1）转换器 A 能把转换器 H 作为先前讨论过的类型 I 的 CGI 程序对待。在这种情况下，转换器 A 能停止监视对话中未来的超文本链接。

（2）转换器 A 能把转换器 H 作为类型 II 的 CGI 程序对待，并继续修改超文本链接。此外，转换器 A 能对未来的 HTML 页面增

加特殊的链接，这将允许用户脱离任何一个转换器的控制。动态修正页面的其他例子

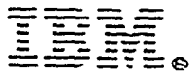
本发明还有这样的特点：它提供系统和方法在对话中客户机浏览 HTML 文件时过滤所有的由客户机观看的 HTML 文本。例如，假定一个客户机已与一个服务器接触并开始一个对话。该服务器希望过滤全部 HTML 文本并去掉已决定排除的短语和超文本链接。本发明提供当客户机访问文件和程序时过滤和/或修改 HTML 文本的方法，这些文件和程序对于服务器进行过滤而言可能是远程的文件和程序。

本发明可应用于那些在对话过程中需要修改 HTML 页面的多种应用中。例如，假定一个服务器应用希望在对话中对传给客户机的全部 HTML 页面进行过滤。转换器能在把 HTML 页面发送给客户机之前修改和/或去掉这些 HTML 页面的不希望的部分。这只要修改转换器使其对不同的子串寻找文本。请注意，转换器能审查存在于远程服务器上的页面和 CGI 程序输出。如果客户机能从服务器下载用 Java 之类语言写的程序。则能在客户机上执行转换器审查工作。

作为另一个例子，假定一个客户机 450 处于一个对话中，那里一些主要公司的名字在文本中频繁出现。运行转换器 416 的服务器 410'能访问主要公司主页 URL 的数据库 475。服务器希望在每次数据库中一个公司的名字出现在 HTML 页面中时便增加超文本链接。例如，每当 IBM 或国际商用机器公司的名字出现在一个 HTML 页面中时，服务器便希望把对超文本链接的参考转换到 IBM 的主页。通过这样做，客户机便能够利用点击得到关于在对话中出现的公司的有用信息。通过修改转换器 416 使其寻找数据库中出现的所有公司名字的 HTML 页面便能完成这一功能。每当找到这样一个名字，便在返回客户机的 HTML 文本中插入对该公司主页的一个超文本链接。在超文本链接遵循远程服务器的情况下，转换器 416 能继续监视对话。如前文指出的那样，如果客户机能从服务器下载由

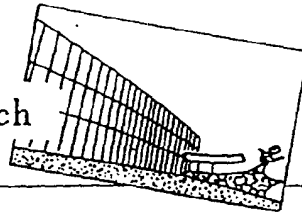
Java 之类语言写成的程序，转换器能在客户机上执行。

既然借助一个最佳实施例和若干变体描述了本发明，对于那些本门技术的专家而言，将能进行各种修改。这样，应该理解，所提供的最佳实施例是作为例子而不是作为限制。所附权利要求恰当地规定了本发明的范围。

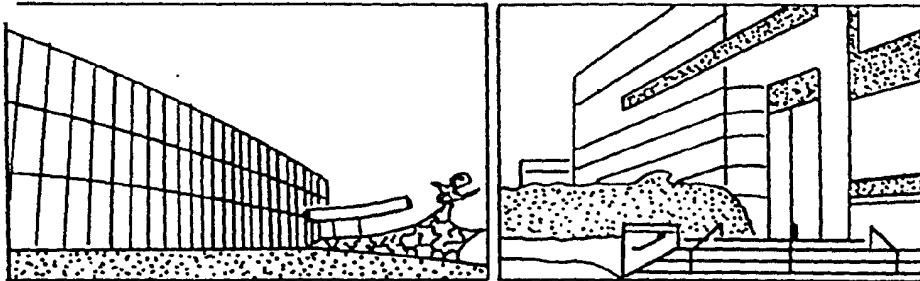


© 1995 IBM Corporation

Research




IBM T.J. Watson Research Center



T.J. Watson Research Center: Yorktown (left) and Hawthorne.

- Welcome!
- Local Education Outreach
- Visitor info and local site directions
- Local hotels
- IBM home page -- IBM Research home page

 Click on icon to send your comments.

Or, contact webmaster@watson.ibm.com

[[IBM home page](#) | [Order](#) | [Search](#) | [Contact IBM](#) | [Help](#) | (C) | (TM)]

图 |
现有技术

图 2

现有技术

天涯供应商登记表

用户标识	<input type="text"/>
口令	<input type="text"/>
口令 (用于确认)	<input type="text"/>
真名	<input type="text"/>
公司	<input type="text"/>
电子邮件地址	<input type="text"/>
电话号码	<input type="text"/>
<input type="button" value="发送"/> <input type="button" value="复位 (表格重置)"/>	

图3

现有技术

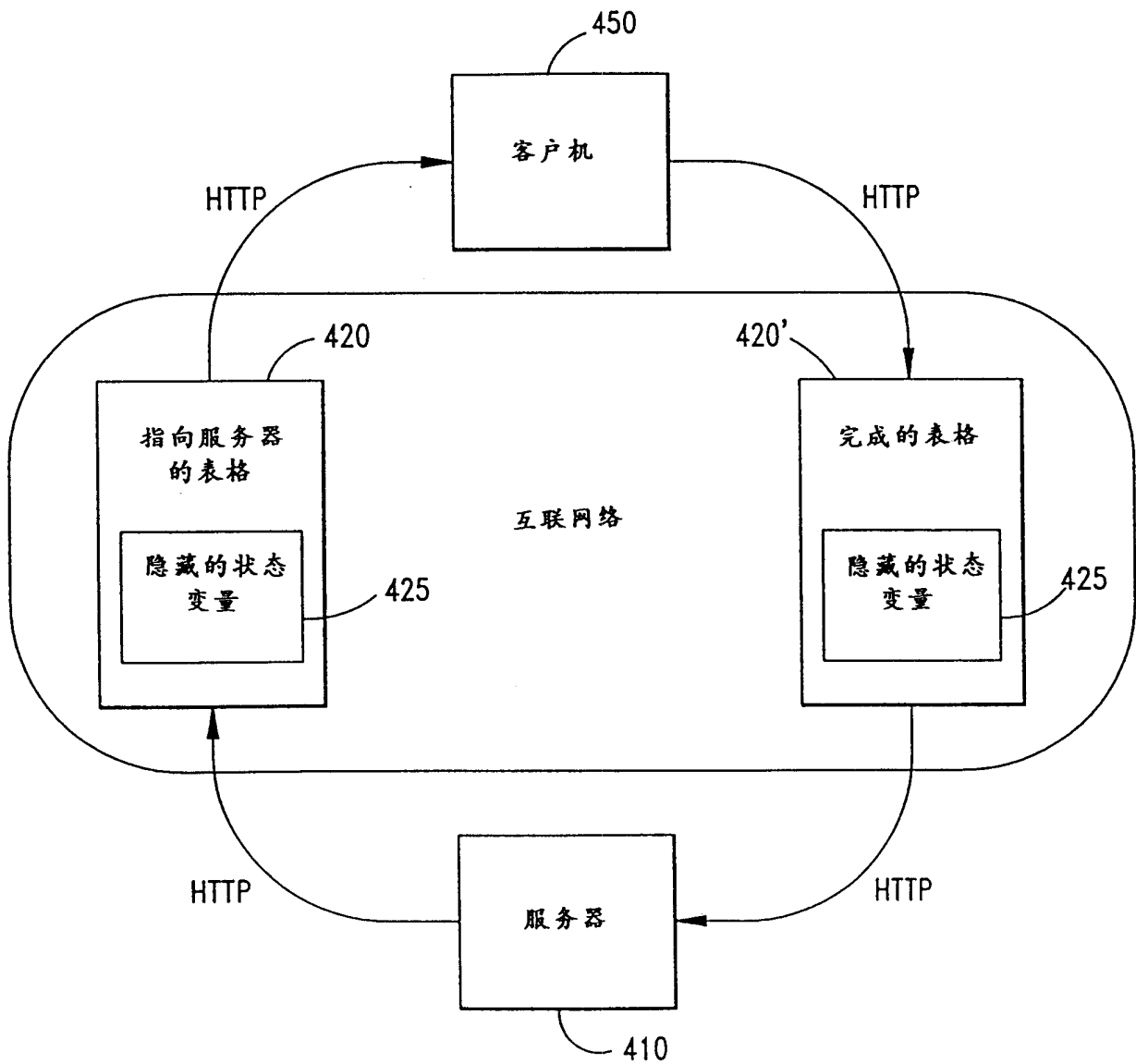
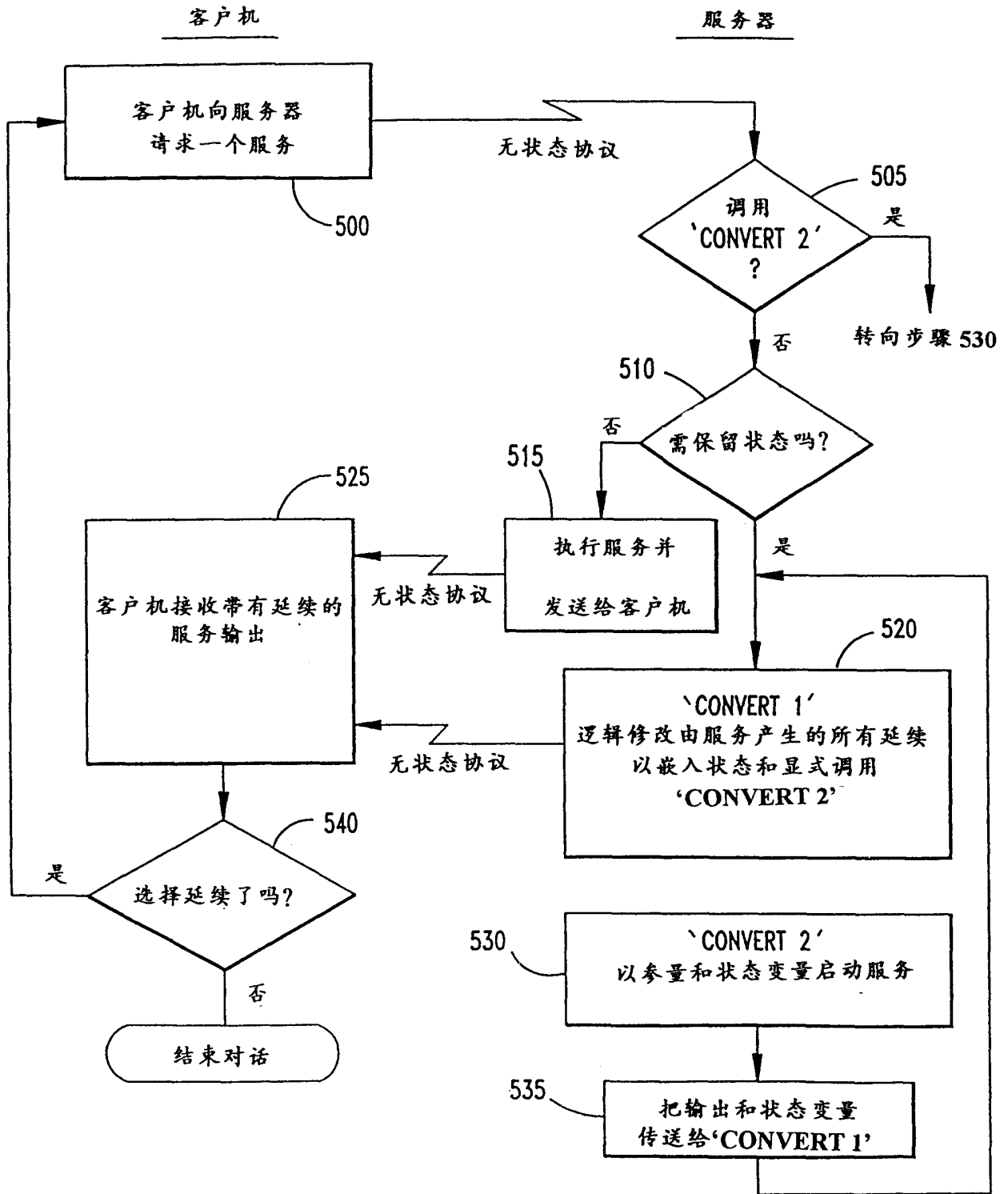


图 4



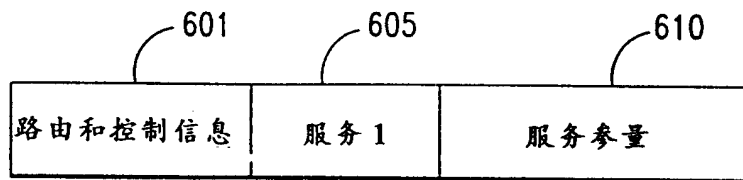


图 5

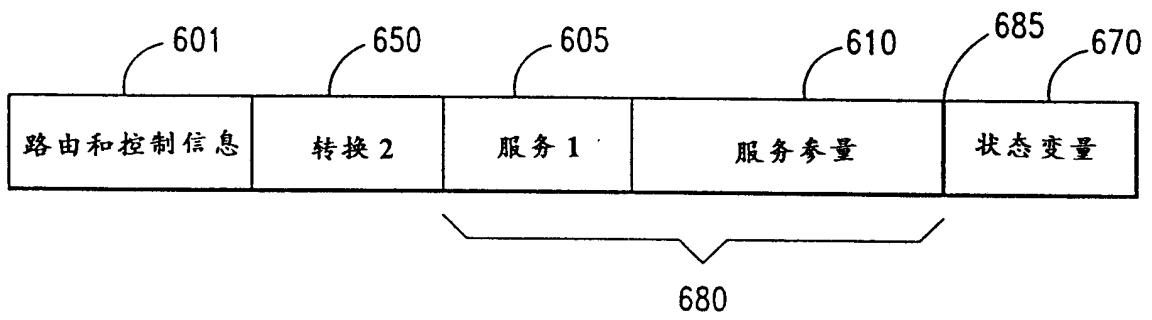


图 6

图 7a

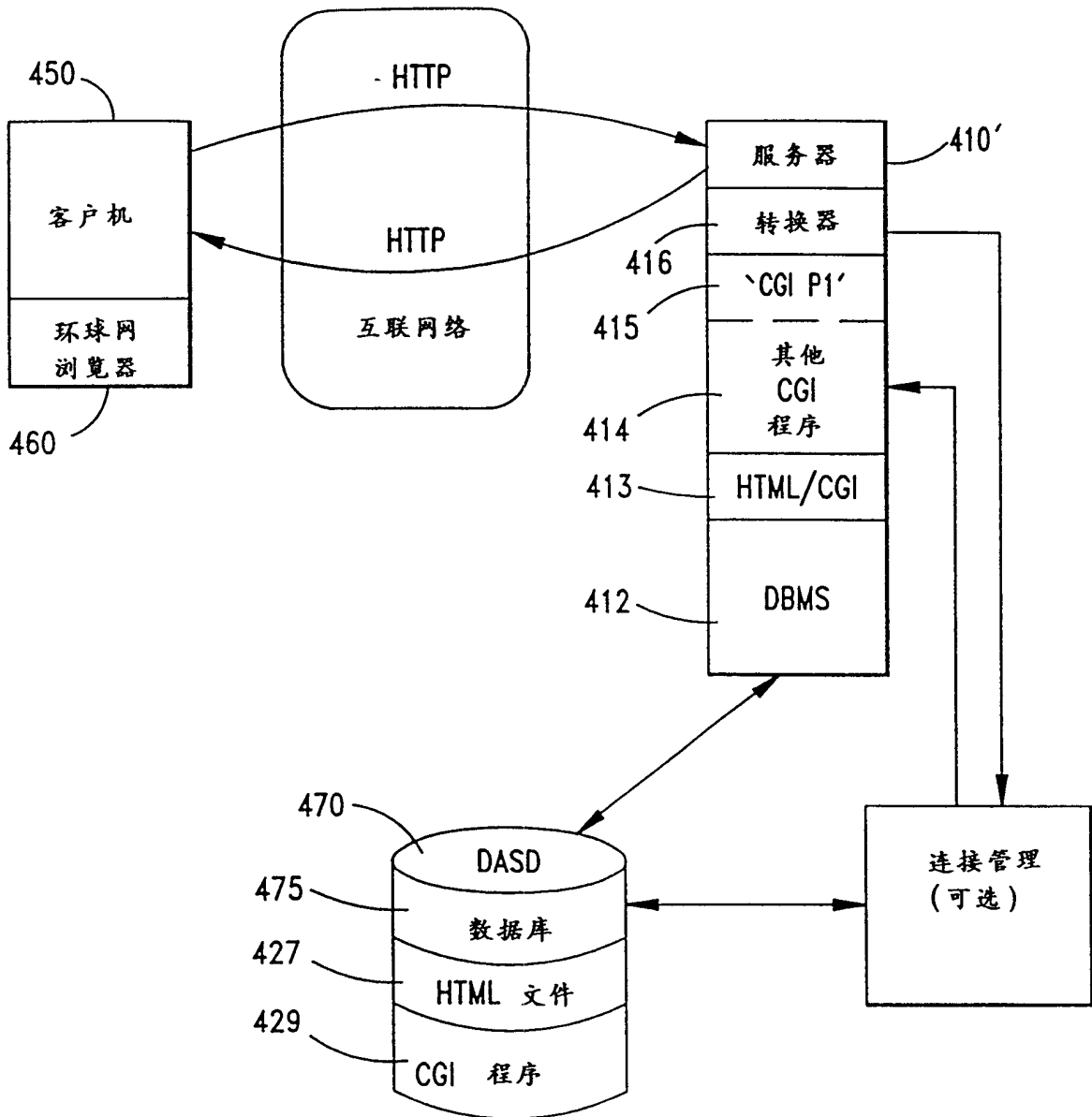


图 7b

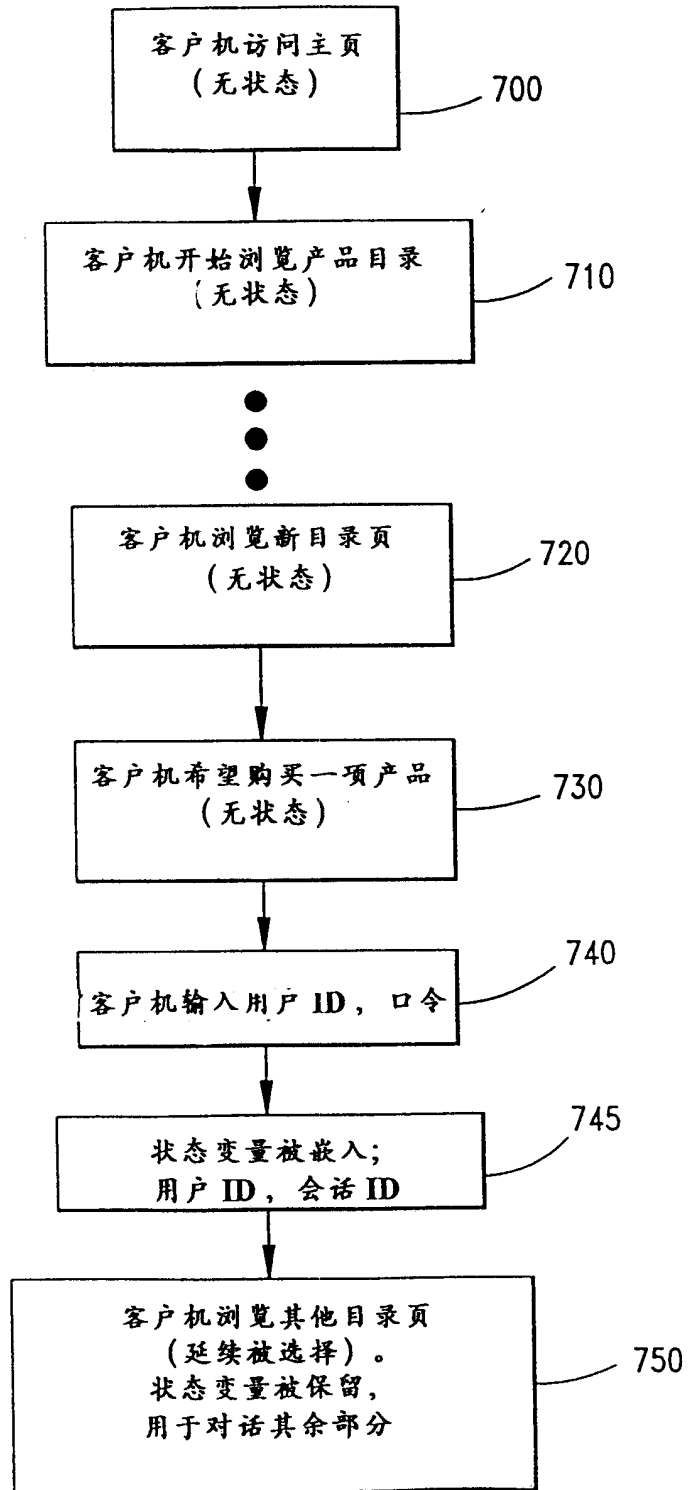


图 8

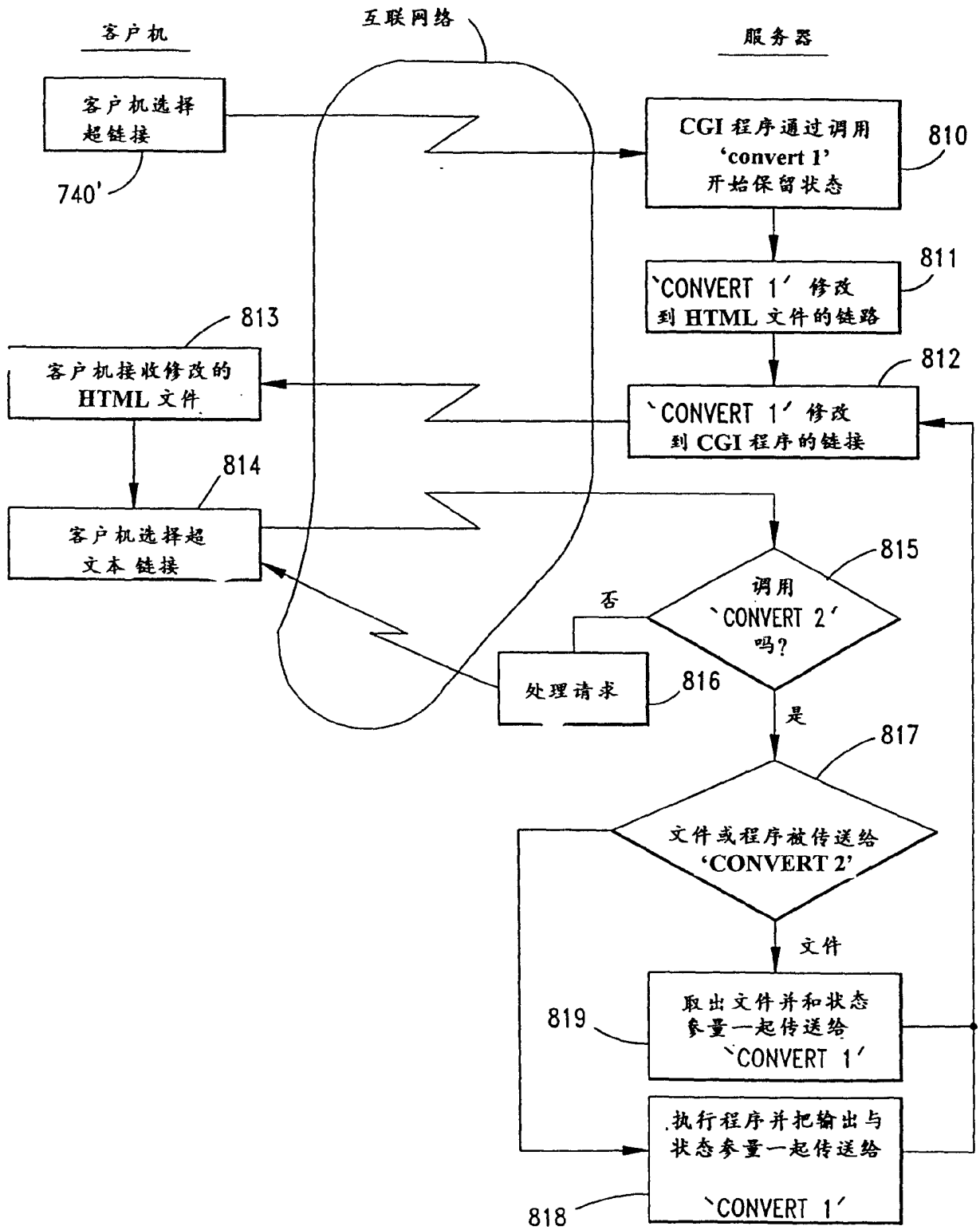


图 9a

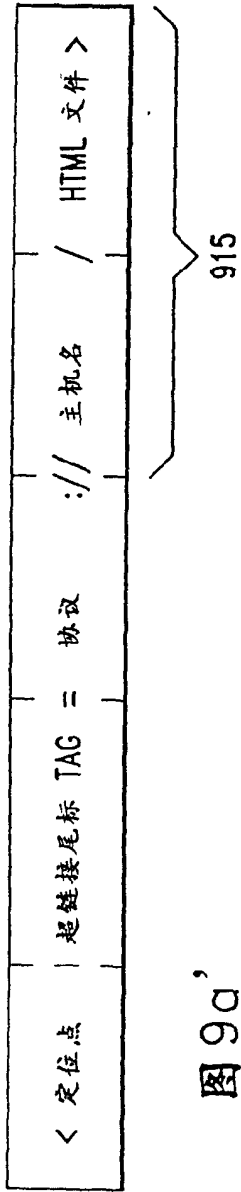


图 9a'

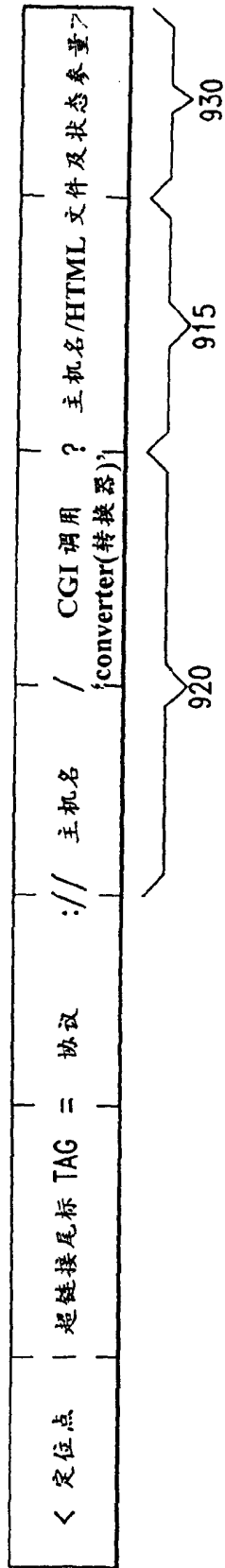


图 9b

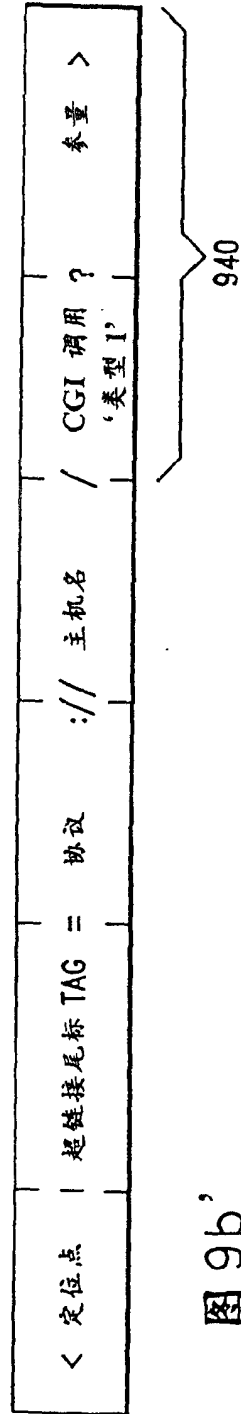


图 9b'

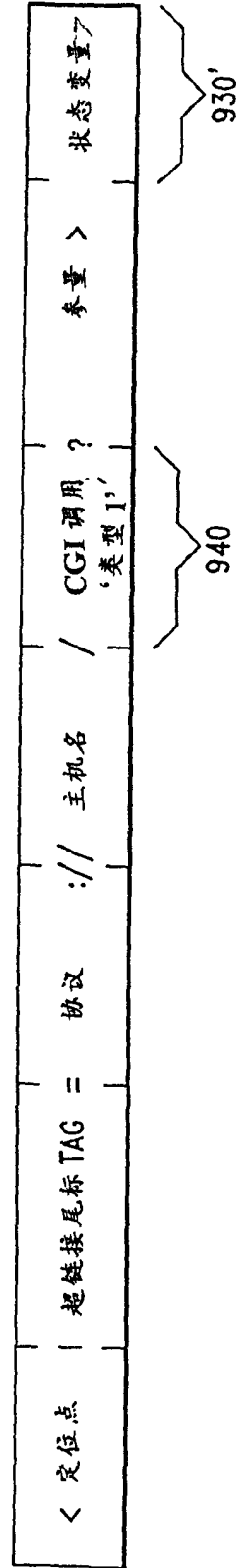


图 9c

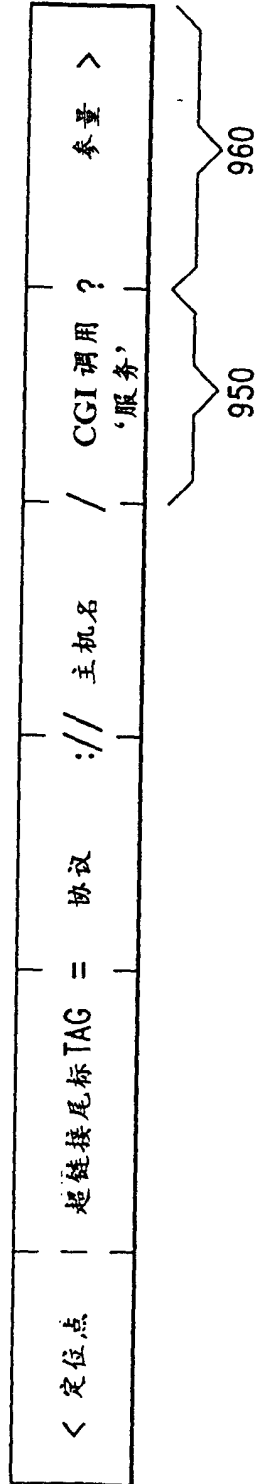


图.9c'

