

[12] 发明专利申请公开说明书

[21] 申请号 99808357.7

[43] 公开日 2001 年 9 月 5 日

[11] 公开号 CN 1311935A

[22] 申请日 1999.5.7 [21] 申请号 99808357.7
 [30] 优先权
 [32] 1998.5.7 [33] US [31] 60/084,578
 [86] 国际申请 PCT/KR99/00222 1999.5.7
 [87] 国际公布 WO99/57838 英 1999.11.11
 [85] 进入国家阶段日期 2001.1.8
 [71] 申请人 三星电子株式会社
 地址 韩国京畿道
 [72] 发明人 理查德·汉普尔曼 王冬岩

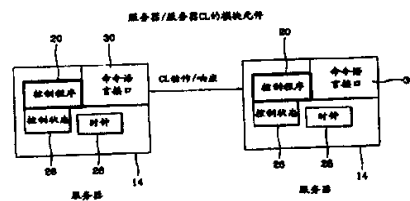
[74] 专利代理机构 柳沈知识产权律师事务所
 代理人 马莹

权利要求书 6 页 说明书 44 页 附图页数 24 页

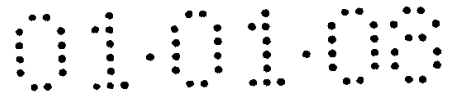
[54] 发明名称 网络中的装置对装置命令与控制方法和系统

[57] 摘要

通过以下方式经由网络在多个装置中进行命令和控制的方法和系统:将第一装置连接到该网络;将第二装置连接到该网络,其中第二装置存储结构化格式的应用程序接口描述数据,用于由其它网络装置命令和控制该第二装置;通过网络将应用程序接口描述数据提供给第一装置;以及利用应用程序接口描述数据将来自第一装置的控制和命令经网络发送到第二装置,以控制第二装置的操作。



ISSN 1008-4274



权 利 要 求 书

1. 一种用于经由网络在多个装置中进行命令和控制的方法，该方法包括下列步骤：

5 (a)将第一装置连接到该网络；

(b)将第二装置连接到该网络，该第二装置存储结构化格式的应用程序接口描述数据，用于由连接到网络的至少一个其它网络装置命令和控制该第二装置；

(c)通过网络将所述应用程序接口描述数据提供给第一装置；以及

10 (d)利用所述应用程序接口描述数据，将来自第一装置的控制和命令数据通过网络发送到第二装置，以控制第二装置的操作。

2. 如权利要求 1 所述的方法，其中步骤(c)还包括：经由该网络定位所述应用程序接口描述数据，并经由该网络将所述应用程序接口描述数据提供给第一装置。

15 3. 如权利要求 1 所述的方法，其中，

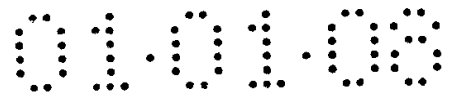
(i)步骤(b)包括：将两个或多个装置连接到该网络，每个装置存储结构化格式的应用程序接口描述数据，用于由连接到网络上的一个或多个其它装置命令和控制该装置；

20 (ii)步骤(c)包括：经由该网络将多个所述装置的应用程序接口描述数据提供给该第一装置

(iii)步骤(d)包括：利用相应于所述多个装置中的每一个的应用程序接口描述数据，将来自第一装置的控制和命令数据，通过网络发送到所述多个装置，以控制所述多个装置的操作。

25 4. 如权利要求 3 所述的方法，其中步骤(ii)还包括经由该网络定位所述应用程序接口描述数据，并经由该网络将所述应用程序接口描述数据提供给第一装置。

30 5. 如权利要求 3 所述的方法，其中步骤(ii)还包括将多个所述装置的应用程序接口描述数据提供给至少第一装置，并且步骤(iii)包括利用相应于所述多个装置中的每一个的应用程序接口描述数据，将来自至少第一装置的控制和命令数据，通过网络发送到连接到该网络的所述多个装置，以控制所述多个装置的操作。



6. 如权利要求 1 所述的方法, 其中步骤(c)包括经由该网络传输至少一部分所述应用程序接口描述数据到该第一家庭装置。

7. 如权利要求 1 所述的方法, 其中步骤(c)包括第一装置经由该网络查询第二装置中的应用程序接口描述数据。

5 8. 如权利要求 1 所述的方法, 其中步骤(c)包括第一装置从连接到该网络的数据库装置中查询应用程序接口描述数据。

9. 如权利要求 1 所述的方法, 其中该应用程序接口描述数据包括关于控制第二家庭装置的操作的第一家庭装置的远程程序调用信息。

10 10. 如权利要求 1 所述的方法, 其中应用程序接口描述数据包括用于识别该第二装置的能力的能力数据。

11. 如权利要求 1 所述的方法, 其中所述装置能够显示用户接口。

12. 如权利要求 1 所述的方法, 其中所述结构化格式包括 XML 格式。

13. 一种用于命令和控制装置的网络系统, 包括:

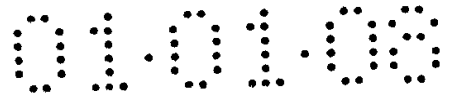
- 15 (a)物理层, 其中物理层提供能够由相互通信的装置使用的通信介质;
- (b)至少一个被控制的装置, 包含结构化格式的应用程序接口描述数据, 用于由至少一个其它装置命令和控制被控制的装置;
- (c)至少一个控制器, 包括控制应用程序部件, 用于获得所述应用程序接口描述数据, 并利用所述应用程序接口描述数据将控制和命令数据发送到被控制的装置, 以控制被控制的装置的操作。

20 14. 如权利要求 13 所述的网络系统, 还包括多个被控制的装置, 每个被控制的装置都存储结构化格式的应用程序接口描述数据, 用于由至少所述控制器命令和控制每个被控制的装置, 其中所述应用程序控制部件有选择地获得一个或多个所述被控制的装置的应用程序接口描述数据, 用于利用所述应用程序接口描述数据, 将控制和命令数据发送到一个或多个所述被控制的

25 装置, 以控制所述一个或多个被控制的装置的操作;

15. 如权利要求 13 所述的网络系统, 其中所述应用程序控制部件通过传输至少一部分所述应用程序接口描述数据到控制器, 来获得所述应用程序接口描述数据, 用于产生和发送所述控制和命令数据给被控制的装置。

30 16. 如权利要求 13 所述的网络系统, 其中所述应用程序控制部件通过在被控制的装置中查询所述应用程序接口描述数据, 来获得所述应用程序接口描述数据。



17. 如权利要求 13 所述的网络系统, 其中所述应用程序控制部件从数据库装置查询所述应用程序接口描述数据, 来获得所述应用程序接口描述数据。

5 18. 如权利要求 13 所述的网络系统, 其中所述应用程序接口描述数据包括关于控制被控制的装置的操作的远程程序调用信息。

19. 如权利要求 13 所述的网络系统, 其中应用程序接口描述数据包括用于识别该被控制的装置的能力的能力数据。

20. 如权利要求 13 所述的网络系统, 其中所述控制器和被控制的装置不能显示用户接口。

10 21. 如权利要求 13 所述的网络系统, 其中所述结构化格式包括 XML 格式。

22. 如权利要求 13 所述的网络系统, 还包括多个控制器, 每个控制器都包括应用程序控制部件, 用于获得一个或多个控制器的应用程序接口描述数据, 用于利用所述应用程序接口描述数据, 将控制和命令信息发送到所述一个或多个被控制的装置, 以控制所述一个或多个被控制的装置的操作。

23. 一种用于经由家庭网络进行服务的方法, 该方法包括步骤:

(a) 将客户机装置连接到家庭网络, 其中该客户机装置能够显示用户接口数据;

20 (b) 将第一家庭装置连接到家庭网络, 该第一家庭装置以选择的格式存储用户接口数据, 用于定义由用户经由该网络对至少第一家庭装置的用户命令和控制的接口;

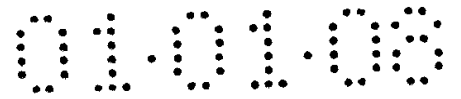
(c) 将第二家庭装置连接到家庭网络, 该第二家庭装置存储结构化格式的应用程序接口描述数据, 用于由连接到该网络的一个或一个以上家庭装置对第二家庭装置的装置命令和控制;

25 (d) 在该客户机装置中经由该家庭网络接收第一家庭装置的用户接口数据;

(e) 在客户机装置上显示由第一家庭装置的用户接口数据定义的用户接口;

30 (f) 响应于与显示在客户机装置上用户接口的用户交互, 从用户处接受用户输入; 以及

(g) 基于该用户输入, 从该客户机装置将控制和命令数据发送到第一家



庭装置，使得该第一和第二家庭装置利用所述应用程序接口描述数据相互通信，以进行服务。

24. 如权利要求 23 所述的方法，其中步骤(f)还包括：从被显示在客户机装置上的用户接口接受用来选择第二家庭装置的用户输入。

5 25. 如权利要求 24 所述的方法，还包括步骤：第一家庭装置基于经由该客户机装置输入到该第一家庭装置的用户输入，利用所述应用程序接口描述数据，通过发送控制和命令信息到第二家庭装置，来控制该第二家庭装置。

26. 如权利要求 25 所述的方法，还包括步骤：经由该网络，将该应用程序接口描述数据提供到第一家庭装置。

10 27. 如权利要求 23 所述的方法，其中所述应用程序接口描述数据包含关于第二家庭装置的能力数据，并且还包括步骤：(i)在第二家庭装置的应用程序接口描述数据中查询该能力数据，以及(ii)利用该能力数据更新第一家庭装置中的所述用户接口数据，以允许用户经由显示在该客户机装置上的第一家庭装置的用户接口来命令和控制第二家庭装置。

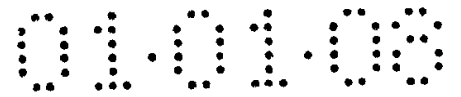
15 28. 如权利要求 23 所述的方法，还包括：连接两个或多个家庭装置到该网络，每个家庭装置存储结构化格式的应用程序接口描述数据，用于由连接在网上的一个或多个其它家庭装置命令和控制该家庭装置。

20 29. 如权利要求 28 所述的方法，其中所述两个或多个家庭装置的每个中的应用程序接口描述数据包含关于各家庭装置的能力数据，并且还包括步骤：(i)在所述两个或多个家庭装置的应用程序接口描述数据中查询该能力数据，以及(ii)利用所述能力数据更新第一家庭装置中的所述用户接口数据，以允许用户经由显示在该客户机装置上的第一家庭装置的用户接口来命令和控制所述两个或多个家庭装置。

25 30. 如权利要求 28 所述的方法，还包括步骤：经由该网络提供多个所述两个或多个家庭装置的应用程序接口描述到第一家庭装置。

31. 如权利要求 30 所述的方法，还包括步骤：经由该网络，利用相应于所述多个家庭装置中的每一个的所述应用程序接口描述数据，从第一家庭装置发送控制和命令数据到所述多个家庭装置，以控制所述多个家庭装置的操作。

30 32. 如权利要求 30 所述的方法，还包括步骤：在该网络上定位所述应用程序接口描述，并经由该网络将所述应用程序接口描述提供到第一家庭装



置。

33. 如权利要求 30 所述的方法，还包括步骤：提供所述多个家庭装置中的一个家庭装置的应用程序接口数据到所述多个家庭装置中的另一个家庭装置。

5 34. 如权利要求 33 所述的方法，还包括：经由该网络，利用相应于所述另一家庭装置的所述应用程序接口描述，从所述一个家庭装置发送控制和命令数据到所述另一家庭装置，以控制所述另一家庭装置的操作。

35. 如权利要求 23 所述的方法，其中应用程序接口描述包括用于第一家庭装置的远程程序调用信息，以控制所述第二家庭装置的操作。

10 36. 如权利要求 35 所述的方法，其中应用程序接口描述包括用于识别第二家庭装置的能力的能力数据。

37. 如权利要求 23 所述的方法，其中被选择的格式包括 HTML 格式。

38. 如权利要求 23 所述的方法，其中该结构化格式包括 XML 格式。

39. 一种用于命令和控制装置的网络系统，包括：

15 (a)物理层，其中物理层提供能够由相互通信的装置使用的通信介质；

(b)第一家庭装置，以结构化格式存储用户接口数据，用于定义对至少第一装置的用户命令和控制的接口；

(c)第二家庭装置，以结构化格式存储应用程序接口描述数据，用于由一个或多个装置对第二服务器的装置命令和控制；

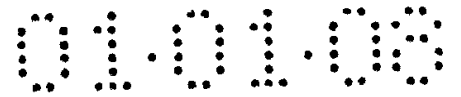
20 (d)客户机装置，能够显示用户接口数据，该客户机装置包括用户接口控制器，用于在该客户机装置上显示该第一服务器的所述用户接口，以接受用户输入，以及基于该用户输入发送控制和命令数据到第一服务器，使得第一和第二家庭装置采用所述应用程序接口描述数据来相互通信，以进行由用户请求的服务。

25 40. 如权利要求 39 所述的网络系统，其中所述用户接口控制器接受用于从被显示在该客户机装置上的用户接口选择第二服务器的用户输入。

41. 如权利要求 40 所述的网络系统，其中第一服务器包括应用程序控制部件，用于基于经由该客户机网络输入到第一服务器的用户输入，利用所述应用程序接口描述数据，通过发送控制和命令信息到第二服务器来控制该

30 第二服务器。

42. 如权利要求 41 所述的网络系统，其中所述应用程序控制部件从该



第一服务器获得该应用程序接口描述数据。

43. 如权利要求 40 所述的网络系统，其中所述应用程序控制部件从一个数据库获得该应用程序接口描述数据。

5 44. 如权利要求 39 所述的网络系统，其中所述应用程序接口描述数据包含关于第二服务器的能力数据，并且应用程序控制部件从该应用程序接口描述数据中获得该能力数据，和利用该能力数据更新第一家庭装置中的所述用户接口数据，以允许用户经由显示在该客户机装置上的第一服务器的用户接口命令和控制第二服务器。

10 45. 如权利要求 39 所述的网络系统，还包括两个或多个服务器，其每个都以所述结构化格式存储应用程序接口描述数据，用于由一个或多个装置命令和控制所述两个或多个服务器。

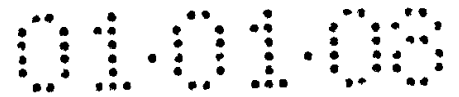
15 46. 如权利要求 45 所述的网络系统，其中所述两个或多个服务器的每个中的应用程序接口数据包含关于各服务装置的能力数据，并且应用程序控制部件从所述两个或多个服务器的应用程序接口描述数据中获得该能力数据，和利用所述能力数据更新第一服务器中的所述用户接口，以允许用户经由显示在该客户机装置上的第一服务器的用户接口来命令和控制所述两个或多个服务器。

20 47. 如权利要求 45 所述的网络系统，其中应用程序控制部件利用相应于所述两个或多个服务器中的每一个的所述应用程序接口描述数据，来发送控制和命令数据到所述两个或多个服务器，以控制所述两个或多个服务器的操作。

48. 如权利要求 39 所述的网络系统，其中所述应用程序接口描述数据包括用于第一家庭装置的远程程序调用信息，以控制所述第二服务器的操作。

25 49. 如权利要求 39 所述的网络系统，其中被选择的格式包括 HTML 格式。

50. 如权利要求 39 所述的网络系统，其中该结构化格式包括 XML 格式。



说明书

网络中的装置对装置命令与控制的方法和系统

5 技术领域

本发明涉及网络系统领域，更具体地说，涉及连接有多个装置的家庭网络(home network)。

背景技术

10 一般来说，网络包括通信链路以及具有与该通信链路连接的通信能力(capability)的各种装置。这些装置包括计算机、外围设备、路由器、存储器以及带有处理器和通信接口的家用产品。网络的一个实例就是用于家庭的家庭网络，在这种网络中互连各种装置。普通的家庭可能拥有包含计算机以及一般家里都找得到的家庭装置在内的几种家庭装置。因而，术语“装置”一
15 般包括逻辑装置或其他具有交换数据功能和能力的器具，它不仅能包括全部家庭装置还包括通用的计算机。家庭装置包括如下电子装置：安全系统、影院系统、TVS、VCR、立体声音响系统，以及也被称为数字卫星服务(DSS)的直接广播卫星服务(DBSS)系统、自动喷水系统、照明系统、微波用具、洗碗机、烤箱/炉、洗衣机/干衣机以及汽车中的处理系统。

20 一般来说，家庭装置用于实现改善主人的生活方式以及提高其生活标准的任务。例如，洗碗机实现洗脏碗的任务，从而将主人从必须用手洗碗中解脱出来。VCR 可以记录电视机目，以使得主人能够在以后的某个时间观看特别的电视机目。安全系统保护主人的财产，并能降低主人对非法闯入的恐惧。

25 象家庭影院这样的家庭装置一般使用单一公用控制器，即遥控器进行控制。这种单一公用控制器允许主人采用单一接口控制和命令几种不同的家庭装置。因此，很多制造商已经开发出用于从单一接口控制和命令他们的装置的控制器。与使用遥控器命令和控制家庭装置有关的一个缺点是提供用于控制和命令每一种家庭装置的静态和命令逻辑。因此，特殊的遥控器只能控制
30 和命令包括有必要的控制和命令逻辑的那些家庭装置。

在传统网络系统中，用户使用遥控器或装置控制面板提供命令。一旦用



户停止操作，则在该网络中不存在任何为自动操作提供命令的控制器单元或装置。例如，家庭可能拥有调谐器装置和不带调谐器的 VCR。在用对 VCR 编程进行延时记录之后，需要一种让 VCR 在以后自动通过网络定位调谐器并控制该调谐器为该 VCR 提供用于记录的视频信息的机制。因而在用户先
5 控制和命令第一组装置后，传统系统不提供必要时第一组装置自动与该网络中的第二组装置通信，以便在不需要第二组装置的直接用户控制和命令的情况下完成任务的机制。

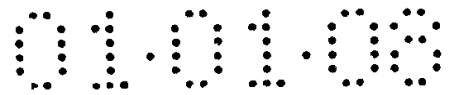
因此需要一种方法和系统，其在家庭网络中提供装置的动态及中央控制和命令。也需要这样的方法和系统，提供用户先控制第一组装置以便相互通
10 信，以及必要时第一组装置与该网络中的第二组装置自动通信，以便在不需要第二组装置的直接用户控制和命令的情况下完成任务的能力。还需要这样的方法和系统，提供各种网络装置自动命令和控制其它各种网络装置的能力。

发明概述

15 本发明满足以上这些需要。在一个实施例中，本发明提供通过以下方式经由网络在多个装置中进行命令和控制的方法和系统：将第一装置连接到该网络；将第二装置连接到该网络，其中第二装置存储结构化格式的应用程序接口描述数据，用于由其它网络装置命令和控制该第二装置；通过网络将应用程序接口描述数据提供给第一装置；以及利用应用程序接口描述数据，将
20 来自第一装置的控制和命令通过网络发送到第二装置，以控制第二装置的操作。

至少一部分应用程序接口描述数据可以传输到第一装置使用，或第一装置可以经由网络查询其它装置中的应用程序接口描述数据。此外，应用程序接口描述数据可以存储在第一装置存取的数据库中。该应用程序接口描述数
25 据可以包括第一家庭装置控制第二家庭装置的操作的远程程序调用信息。此外，该应用程序接口描述数据包括识别第二装置的能力的能力数据。用于应用程序接口描述数据的结构化格式可以包括 XML 格式。

最好，在网上连接多个装置，并且每个装置包含结构化格式的应用程序接口描述数据，用于由一个或多个连接到网络上的其它装置命令和控制该装
30 置。此时，两个或两个以上装置的应用程序接口描述数据可以经由该网络提供到第一装置，因此，第一装置可以利用相应于两个或两个以上装置的每一



个的应用程序接口描述数据，将来自第一装置的控制和命令数据通过网络发送到第二装置，以控制它们的操作。

另一方面，本发明提供通过以下方式经由家庭网络进行服务的方法和系统：将客户机装置连接到家庭网络，其中该客户机装置能够显示用户接口数据；将第一家庭装置连接到网络，其中第一家庭装置包含有选定格式的用户接口数据，用于定义至少第一家庭装置的用户命令和控制的接口；将第二家庭装置连接到网络，其中第二家庭装置包含有结构化格式的应用程序接口描述数据，用于第二家庭装置的装置命令和控制；在客户机装置上接收第一家庭装置的用户接口数据；在客户机装置上显示由第一家庭装置的用户接口数据定义的用户接口；响应于与显示在客户机装置上的用户接口的用户交互，接受用户的输入；以及基于该用户输入将来自该客户机装置的控制和命令数据发送到第一家庭装置，使得第一和第二家庭装置利用应用程序接口描述数据相互通信，以进行服务。

因而，用户可以从在客户机装置上显示的用户接口中选择第一和第二家庭装置。此外，第一家庭装置可以通过利用基于经由客户机装置用户输入到第一家庭装置的应用程序接口描述数据，向第二家庭装置发送控制和命令信息，来控制第二家庭装置。该应用程序接口描述数据可以包括关于第二家庭装置的能力数据，并且该第一家庭装置可以在第二家庭装置的应用程序接口描述数据内查询该能力数据，以及更新第一家庭装置中的用户接口数据，以允许用户经由显示在客户机装置上的第一家庭装置的用户接口来命令和控制第二家庭装置。

两个或两个以上家庭装置可以连接到网上，每个家庭装置都存储结构化格式的应用程序接口描述数据，用于由连接到网络的其它家庭装置命令和控制该家庭装置。该两个或两个以上家庭装置的每个中的应用程序接口描述数据可以包括各家庭装置的能力数据。在这种情况下，第一家庭装置可以在该两个或两个以上家庭装置的应用程序接口数据内查询该能力数据，以及更新第一家庭装置中的用户接口数据、以允许用户经由显示在客户机装置上的第一家庭装置的用户接口来命令和控制该两个或两个以上家庭装置。

附图简述

根据下列的描述、所附权利要求和附图，本发明的这些以及其它特性、特征和优点将更加容易被理解，其中：

图 1 是根据本发明的一个方面的网络的实施例的方框图；

图 2 是示例性装置控制和通信情况(scenario)中的图 1 的方框图；

图 3 是根据本发明的、包括多个客户机和服务器的一个示例性家庭网络系统的方框图；

5 图 4 是图 3 的一个客户机和一个服务器的示例性实施例的方框图；

图 5 是客户机装置的示例性实施例；

图 6 是服务器(server device)的示例性实施例；

图 7 是两个能够相互通信并且控制的联网服务器的方框图；

10 图 8 是在一个网络中包括一个信源(source)服务器、一个信宿(sink)服务器以及客户机装置在内的示例性音频/视频(A/V)模型的示例性结构方框图；

图 9 是另外一个示例性音频/视频(A/V)模型；

图 10 是一个网络装置的示例性能力数据表；

图 11 是一个网络装置的示例性属性数据表；

图 12 是用于在联网装置之间产生命令消息的构件块的示例性配置；

15 图 13 是用于产生命令消息的图 12 的构件块的另一个示例性配置；

图 14 是联网的客户机装置和服务器之间的交互的三个实例；

图 15 是关于联网的装置接口的 API 扩展的定义的示例性方框图；

图 16 是关于接入另一个服务器的接口描述文档的一个服务器应用程序的示例性结构；

20 图 17 是控制服务器和受控服务器之间的另一个示例性装置间(inter-device)控制体系机构；

图 18 是在联网的装置之间的 API 级的通信栈中提供 Web(网络)标准公共中间件层的 XML 协议的实施例；

图 19 是服务器对服务器命令和控制体系结构的另一个实施例；

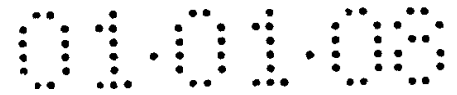
25 图 20 是装置接口库和用于家庭网络的家用电器定义数据库之间的关系；

图 21 是装置接口定义的实施例的分层形式；

图 22 是图 21 的装置接口定义中的层的示例；

图 23 是发送器和接收器之间的命令传输和解释过程；以及

30 图 24 是根据本发明的一个方面、提供翻译服务的信息包类型和格式的部分清单的示例性表。



发明的详细描述

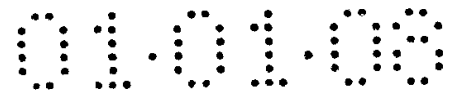
一方面，本发明在诸如家庭网络的网络中提供装置间通信。由于家庭装置越来越智能化并且可以共享信息，所以装置间通信允许用户互连网络中的装置，以利用共享那些装置的能力的信息。因而，装置间通信在给予用户全面便捷地使用联网的装置方面起了主要作用。

参看图 1，在本发明的实施例中，网络 10 包括至少一个客户机装置 12 和经由通信链路 16 互连的至少一个服务器 14。该通信链路 16 可以包括提供用于在各种被连接的家庭装置之间发送和接收数据的物理层(介质)的 1394 串行总线。该 1394 串行总线既支持时分多路复用音频/视频(A/V)流，也支持标准的 IP(互联网协议)通信。在某些实施例中，家庭网络将 IP 网络层作为家庭网络的通信层使用。但可使用其它通信协议为家庭网络提供通信。

每个客户机装置 12 可以与网络 10 中的一个或多个服务器 14 通信。此外，每个服务器 14 也可以与网络 10 中的一个或多个其它的服务器 14 以及一个或多个客户机装置 12 通信。每个客户机装置 12 可以包括含有象用于接收用户输入的鼠标和键盘的输入装置的通信接口、以及用于为用户提供控制用户接口以便于用户与联网的装置进行交互的显示器。该用户接口可以包括一个图形用户界面(GUI)显示器 18，用于给用户的信息。参考图 2，正如这里限定的那样，每个服务器 14 除提供控制用户接口外，还为用户提供一种服务，而每个客户机装置 12 为用户提供与网络 10 交互的控制用户接口。因而，只有客户机装置 12 直接与用户交互，服务器 14 仅仅与客户机装置 12 和其它服务器 14 交互。示例性服务可以包括 MPEG 的发布(sourcing)/汇集(sinking)以及显示服务。

图 3 是包括多个客户机装置 12 和多个服务器 14 的一个示例性家庭网络的框图。每个服务器 14 可以包括作为网络资源的硬件，用于给用户提供服务。此外，每个服务器 14 可以存储用于控制该服务器硬件的服务器或服务控制程序 20，并且可以包括图形控制对象(GCO)用户接口描述 22，作为用户与服务器控制程序 20 交互的接口，如图 4 所示。

为了控制的客户机装置 12 和被控制的服务器 14 之间的控制，客户机装置 12，例如，通过在网络上从服务器 14 传输 GCO 22 到客户机装置 12 来接入服务器 14 的 GCO 22。然后客户机装置 12 使用被传输的 GCO 22 为用



户创建控制用户接口 GUI 18，以便于在该网络上从客户机装置 12 与服务器 14 的服务器控制程序 20 通信。用户从客户机装置 12 给至少服务器 14 的服务器控制程序 20 提供命令和控制。

5 在服务器自身中存储的每个服务器 14 的 GCO 22 可以降低带有几个服务器 14 的网络中客户机装置 12 的处理和存储要求。此外，在服务器 14 中的存储 GCO 22 可以允许每个服务器 14 提供自己的 GUI 外观和感觉给用户，而且也允许在不修改客户机装置 12 的前提下，修改或更新 GCO 22。

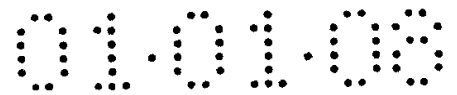
参考图 4，为了在客户机装置 12 和服务器 14 之间提供命令和控制，在一个实施例中包括一个呈现器(renderer)24，用于利用存储在客户机装置 12 10 中、或在该网络上被从所需的服务器 14 中传输到客户机装置 12 的 GCO 22，显示 GUI 18。例如，在初始装置选择阶段，客户机装置 12 可以在该网络上取回至少一个服务器 14 的 GCO 22，然后呈现器 24 利用该 GCO 22 显示 GUI 18，以控制服务器 14。该 GUI 18 最好针对服务器 14 进行定制，并且最好能够包括用于控制服务器 14 的内置命令集。

15 此外，各服务器 14 的 GCO 22 可以包括如下共同点：(1)用于显示 GUI 18 的客户机装置呈现器 24 的共用 GCO 模型，(2)用于将 GCO 22 从服务器 14 传输到客户机装置 12 的公共通信协议，以及(3)用于从客户机装置 12 到服务器 14 的服务器控制程序 20 的 GUI 交互的公共通信协议，其中客户机装置 12 不需要被控制的特别服务器 14 的内置知识(built-in knowledge)。

20 再参考图 4，服务器 14 可以包括一个或多个服务器控制程序 20，以便控制用于提供服务的服务器硬件。来自服务器 14 的 GCO 22 的 GUI 接口 18 为服务器控制程序 20 提供接口。该服务器 14 还可以包括在提供的请求服务时指示该服务器 14 和服务器硬件的控制状态的控制状态数据 26。

25 例如，控制状态数据 26 可以包括服务器 14 的 GUI 18 中的控制信息状态，如 VCR 服务器中记录动作的定时器设置。为了服务器 14 的用户控制，控制状态数据 26 存储在被控制的服务器 14 中，并在控制客户机装置 12 上通过服务器 14 的 GUI 18 显示给用户。用于显示服务器 14 的 GUI 18 的控制客户机装置 12 不保持被控制的服务器 14 的控制状态数据 26 的知识。

30 每个服务器 14 可以由一个或多个客户机装置 12 进行控制。因而，存储在服务器 14 中的控制状态数据 26 包括在每个控制客户机装置 12 中服务器 14 的 GUI 18 中的信息的状态。例如，当用户采用第一客户机装置 12 控制

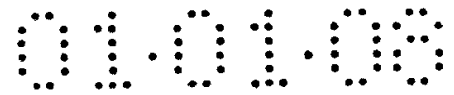


服务器 14 时，该用户控制结束后，在第一客户机装置 12 上服务器 14 的 GUI 18 中的信息也被服务器 14 保存在该服务器 14 的控制状态数据 26 中。

另外，当用户在第一客户机装置 12 上与服务器 14 的 GUI 18 进行交互的同时，将服务器 14 的控制状态数据 26 用在客户机装置 12 上服务器 14 的 GUI 18 中的信息进行更新，并且在用户控制结束后，控制状态数据 26 保留在服务器 14 中。当用户采用第二客户机装置 12 控制服务器 14 时，经由在第二客户机装置 12 上服务器 14 的 GUI 18，使控制状态数据 26 成为用户进一步的控制之用。用户也可以在以后使用第一客户机装置 12 控制服务器 14，因此经由在第一客户机装置 12 上服务器 14 的 GUI 18，也能使控制状态数据 26 成为用户进一步的控制之用。如下所述那样，服务器 14 还可以包括时钟 28，或维持当前时间，以便根据时间或用户的时钟输入允许时间延迟动作。

客户机装置 12 和服务器 14 可以物理地作为象 DTV 那样的一个单元捆绑(bundle)在一起。此时，服务器 14 包括用于控制服务器硬件的控制程序 20，并且客户机装置 12 提供控制用户接口给服务器控制程序 20，用于至少该服务器硬件的控制和命令。图 5 是客户机装置 12 的一个实例，它可包括：(1)用于显示 GUI 的 PDA(RemoteC)，(2)用于显示 GUI 并包含包括音频和/或视频节目条纹(streak)目标服务器的信宿服务器的 DTV(STB)，以及(3)用于显示 GUI 并包括用于提供多种服务的至少一个服务器的 PC。DTV 或 PC 客户机装置中的硬件和可执行部分还可以由其它客户机装置控制。图 6 是示例性的服务器 14，它包括：(1)作为信源服务器的 DVDP 智能卡，(2)作为信宿服务器的音频放大器，(3)作为信源或信宿服务器的 DVCR，以及(4)用于管理远端服务器的管理服务器。该管理服务器可以包括在例如 DBS-STB、有线 TV-STB 或 ATSC-STB。这样的装置包括适合于该 STB 的内部运行的本地控制或管理的管理服务器。此外，通过外部网络接入的外部服务器可以由本地客户机装置用于服务，诸如视频点播、增强电视(Enhanced-TV)、互联网商务之类。

参考图 7，两个服务器 14 之间的通信和控制由在其间传输命令和控制数据的服务器 14 的控制程序 20 实现。服务器 14 可以控制一个或多个在该网络上的其它服务器 14。并且服务器 14 也可以由一个或多个服务器 14，以及一个或多个客户机装置 12 控制。此外，用户可以利用客户机装置 12 控制



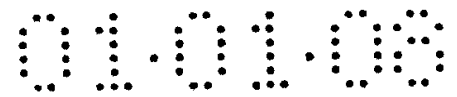
和命令第一组服务器 14，而该第一组服务器 14 在需要为用户执行服务时，可以自动命令和控制第二组服务器 14，而不用用户的干预。

5 例如，对于自动时间延迟操作，用户能够“登录”到客户机装置 12，以便控制第一组服务器 14 并指定想要的服务。然后用户从客户机装置 12 中“退出”。第一组服务器 14 在它们自身之间执行通信和控制，并在以后，该第一组中的一个或多个服务器 14 在需要集中提供所需服务时，自动地控制第二组服务器 14 而不用用户的干预。

10 图 7 是两个能够相互通信并相互控制的服务器 14 的示例性实施例。每个服务器 14 都包括上述的控制程序 20、时钟 28 和控制状态数据 26。每个服务器 14 还可以包括由客户机装置 12 直接控制的服务器 14 的 GCO 22。但 GCO 22 不需要包括在不直接由客户机装置 12 控制的服务器 14 中，并且只与其它服务器 14 通信。每个服务器 14 还包括一种命令语言(CL)接口 30 以及一个命令库。该命令库包括服务器 14 用来发送和接收用于提供其服务的信息的命令。但对于图 4 所示以及以上所述的用户控制，命令语言不是必要的。
15

20 图 8 是在网络上包括信源服务器 14、信宿服务器 14 和客户机装置 12 的示例性音频/视频(A/V)模型。信源服务器 14 包括用于控制该信源服务器 14 的数据流信源硬件 32 的服务器控制程序 20。该信宿服务器 14 包括用于控制信宿服务器 14 的数据流信宿硬件 34 的服务器控制程序 20。在一个示例性操作中，用户利用客户机装置 12 控制信源服务器 14 来启动数据流信源硬件 32，并控制信宿服务器 14 来启动数据流信宿硬件 34。在从数据流信源硬件 32 到数据流信宿硬件 34 的数据传输起动时，用户可以释放客户机装置 12。另外，用户可以安排在以后的时间起动数据传输，并释放客户机装置 12。此后，信源服务器 14 的数据流信源硬件 32 和信宿服务器 14 的数据流信宿
25 硬件 34 在由用户安排的时间自动起动该数据传输。

30 例如，数据流信源硬件 32 可以包括诸如直接广播卫星(Direct Broadcast Satellite, DBS)这样的调谐接入装置(Tuner-Access Device)。DBS 是替代有线电视的多通道，并且直接地从卫星提供类似有线电视的节目到小(直径在 18 英寸至 3 英尺)卫星抛物面天线上。借助于 DBS，几路标准的模拟电视信号被数字地压缩到单一卫星转发器上，以允许多达 200 或更多路由指向空中的固定位置的抛物面天线的通道被接收到。数据流信宿硬件 34 可以包括数



字磁带录像机(DVCR)，该数字磁带录像机含有能够在播放时解压缩数字视频信号的数字 VCR。用户提供包括用于 DVCR 的“延时记录”事件数据和用于调谐接入装置的“延时选择节目”事件数据。在该延时之后，调谐接入装置选择想要的节目，以及源节目数据给接收和记录该节目数据的 DVCR，
5 而不用来自用户的进一步控制动作。

图 9 是在网络 10 上包括至少一个信源服务器 14 SERVER 1、一个信宿服务器 14 SERVER 2 和客户机装置 12 的示例性音频/视频(A/V)模型。该客户机装置 12 包括一个带有用户接口的会话管理器(session manager)36，用于显示选择信息，以便用户选择和控制服务器 14 SERVER 1、SERVER 2 以及
10 其它诸如 SERVER 3 和 SERVER 4(未显示在图中)这样的服务器 14。该选择信息可以包括在会话管理器 36 中被标示作 Serv 1、Serv 2、Serv 3 和 Serv 4 的图标符号，以便于用户分别选择服务器 14 SERVER 1、SERVER 2、SERVER 3 和 SERVER 4。信源服务器 14 SERVER 1 可以包括 DVCR，而信宿服务器 14 SERVER 2 可以包括 1/2 DTV。

15 在一个示例性操作中，客户机装置 12 在选择服务器 14 SERVER 1 和 SERVER 2 后传输每个服务器 14 的 GCO 22 给客户机装置，并且显示服务器 14 SERVER 1 和 SERVER 2 中的每个的相应 GUI 18。用户可以与每个服务器 14 的 GUI 18 交互，以便提供命令和控制给相应的服务器 14 进行服务。每个服务器 14 可以单独或和其它服务器 14 组合起来提供服务。此外，当相
20 应的服务器 14 需要执行服务时，会话管理器 36 在客户机装置 12 中服务器 14 的多个 GUI 18 之间传输控制状态数据 26。根据用户命令和控制信息，两个或多个服务器 14 可以在它们自身之间传递命令和控制，以便提供用户请求的服务。

会话管理器 36 包括一个软件代理，它的功能是接入和显示由网络 10 的
25 各服务器 14 提供可用的家庭网络服务。软件代理还可以匹配该网络 10 中的各服务器 14 的能力，并只为具有兼容能力的那些服务器 14 显示选择信息。此外，会话管理器 36 还可以将在一个服务器 14 的 GUI 18 中产生的选择与另一个服务器 14 的 GUI 18 中产生的选择匹配，以帮助用户提供有意义的命令和控制信息给该服务器 14。

30 在一个示例性操作中，会话管理器 36 执行用于搜索该网络并发现连接到该网络的服务器 14 的软件代理。该软件代理也存取存储在每个服务器 14



中的能力数据以判定该服务器 14 的能力，并将关于那些能力的信息提供给用户。然后会话管理器 36 显示关于服务器 SERVER 1、SERVER 2、SERVER 3 和 SERVER 4 的图标 Serv 1、Serv 2、Serv 3 和 Serv 4，如图 9 所示。

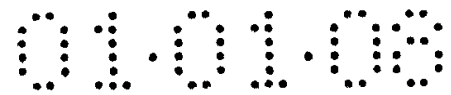
5 会话管理器 36 首先启动全部选择图标 Serv 1、Serv 2、Serv 3 和 Serv 4，以允许用户从全部四个选择图标中选择。在用户通过在 Serv 1 选择图标上点击来选中服务器 SERVER 1 之后，会话管理器 36 判定服务器 SERVER 3 和 SERVER 4 与服务器 SERVER 1 在能力上不兼容。因而，会话管理器 36 分别禁止关于服务器 SERVER 3 和 SERVER 4 的选择图标 Serv 3 和 Serv 4。然后用户在图标 Serv 2 上点击来命令和控制服务器 SERVER 2。

10 当用户与选中的服务器 14 的 GUI 18 交互时，由用户输入到每个 GUI 18 的控制和命令信息提供影响由用户进行的进一步的服务器选择的额外能力信息。例如，如果 VCR 服务器 14 被选中，则会话管理器 36 在启动和禁止其它服务器 14 的选择图标的进一步动作受用户播放或记录的决定的影响。

在该网络中的每个服务器 14 具有一种或多种如以上通过示例、参考图 15 9 所示的服务器所讨论的服务能力。每种服务能力包括信息的发布/汇集。例如，电视具有接收视频和音频流的汇集能力。VCR 装置可以发布(发送)和汇集(接收)视频和音频信号，而 PC 能够发送和接收视频、音频和数据。每种发布能力具有一种配套以及兼容的汇集能力。同样地，每一种汇集能力也具有配套以及兼容的发布能力。例如，一种装置的视频输出能力由另一装置的视频输入能力配套。

20 因为每个装置 14 可以是该网络上的几种不同服务的信源或信宿，因此，每个装置 14 存储一个如图 10 中示范的能力数据表(能力表 1)。表 1 的第一列识别装置 14 的服务能力，第二列识别该装置 14 是否是第一列中相应的服务的信源和信宿。利用该能力数据表 1，可以在维持与旧装置兼容的同时实施新的服务。例如，如果开发了一种与旧服务兼容的新服务，则新旧两种服务都可以进入关于实施该新服务的装置的能力数据表 1，从而实施的装置仍然保留与使用该旧服务的较旧的装置的兼容性。

30 在一种实施中，装置管理器(Device Manager)对装置信源和信宿服务进行匹配或比较。例如，该装置管理器可以实施为软件代理，以比较各装置 14 的能力或性能并用匹配的能力定位装置 14。例如，在该装置为从第一装置 14 通过该网络到第二装置 14 的一个介质流的情况下，该装置管理器比较该第



一和第二装置 14 的能力，以便协助用户明智地选择兼容于第一装置 14 的能力的第二装置 14。以下是关于服务器 14 实施例的服务能力的示例性清单：

Stream_format_video_dv

Stream_format_video_mpeg2tpt

5 Stream_format_video_dsstpt

Stream_format_video_mpeg2pes

Stream_format_video_mpeg210801-tpt

10 每个装置 14 还可以存储一个包含该装置的相关属性的属性数据表(属性表 2)，该表被示范在图 11 中。名称和值判定了表 2 中的每一属性。虽在表 2 中显示了字符长度，但这不是必要的。该属性数据可用于该网络 10 上的其它装置 14，以便于协作和存储装置信息。例如象以下描述的装置页面(Device Page)使用该属性表 2 存储该装置名称。其它字段在必要时可以加进该属性数据表 2。

15 在以上描述的用户-客户机(user-to-client)装置控制模式中，属性数据可以显示与该客户机装置 12 中的服务器 14 的 GUI 页面上。另外，第二级装置信息主页面可以用来显示所述属性数据。此外，表示成文本形式或可扩展标记语言(XML)文档形式的属性数据可以由软件代理存取。对于装置-装置(device-to-device)控制模式，用于该被控制的装置的属性数据存储在该装置接口应用程序接口(device interface application interface)中。

20 在属性表 2 中的装置位置属性字段用于存储每个装置 14 的位置或组。该装置类型属性字段指定装置的类型，例如对具体的装置指定其类型 VCR、DVD、DTV、便携式摄像机、PC、安全系统等。该装置类型属性字段在装置本身不提供图标时，用于选择一个缺省装置图标代表该装置页面中的装置。该属性表 2 可以包括用于缺省信源和缺省信宿属性字段的多个项目。对于由装置 14 管理的每一数据类型，每个这样的项代表一个不同的缺省信源或信宿装置 14。

25 该能力和属性数据最好采用分层语言(hierarchical language)封装为结构化数据。这提供了检索用于诸如在 GCO 传输和服务器-服务器(server device-to-server device)控制中其它目的的能力和属性数据的一种通用方法。例如，该属性数据可以包括下列结构化数据格式：

<DEVICEATTRIBUTES>

```

<ATTRIBUTE name=DeviceManufacturer value= "Samsung Inc." >
<ATTRIBUTE name=Manufacturer URL value=www.samsung.com>
<ATTRIBUTE name=ManufacturerIcon value= "logo.gif" >
<ATTRIBUTE name=DeviceName value= "Samsung DSS" >
5 <ATTRIBUTE name=DeviceModel value= "SCH1900" >
<ATTRIBUTE name=DeviceType value=DDS>
<ATTRIBUTE name=DeviceLocation value= "Livingroom" >
<ATTRIBUTE name=Deviceicon value= "device.gif" >
<ATTRIBUTE name=DeviceAddress value=105.144.30.17>

```

10 </DEVICEATTRIBUTES>

例如，能力数据可以包括下列结构化格式：

```

<DEVICECAPABILITIES>
  <CAPABILITY type=MPEG2 value=Source>
  <CAPABILITY type=MPEG2 value=Sink>
15 <CAPABILITY type=MPEG3 value=Source>
  <CAPABILITY type=MPEG3 value=Sink>

```

</DEVICECAPABILITIES>

应用程序接口语言用于使不同服务器 14 能够执行包括服务器-服务器控制的装置-装置控制。该应用程序接口语言包括命令语言，并且可以采用 XML 描述，细节如下。一个服务器 14 的控制程序 20 远程地控制在该网络上的另一服务器 14 的控制程序 20，而不使用 GUI 18 或不必要用户介入。装置-装置控制的一个示例是自动操作。用户首先为想要的服务通过客户机装置 12 提供控制，然后两个或两个以上的服务器 14 自动相互通信和控制以便提供该服务，而不用用户进一步的交互。

25 参考图 12 和 13，标准的应用程序接口语言最好用于允许在各服务器 14 中的各控制程序 20 之间的协作。在一个实施例中，该标准应用程序接口语言包括下列构造部件(building block)：(1)诸如服务函数数据库中的服务的功能说明 40，(2)在其中包括消息单元的块 42，(3)行业标准格式 44，(4)消息压缩 46，(5)输出结构化的消息数据的消息串结构 48。

30 图 12 是执行产生命令消息的函数的构造部件的示例性结构。每一消息项目由服务的函数规格说明组成，并通过选择关于该消息项的行业标准化压



缩的格式(Hex)标签来标准化。这样的消息项组汇集在一起产生完整的命令串。诸如 CAL 和 AV/C 的现有命令语言按图 12 所示那样操作。但这样的命令语言机构规定了物理接口上的物理装置上的二进制或十六进制代码消息和系统操作，并且是基于硬件规格的。因此，这样的命令语言对于基于网络层的控制机构来说不是很必要，即在該机构中控制系统规格说明包括应用程序级的软件级的命名、寻址、装置能力发现、通信语言和命令消息，其中在控制器 14 的一个软件应用程序 20 定位和控制在网络 10 上被控制的装置 14 中的另一个软件应用程序 20。所述控制结构更适合于诸如包括家用电器(例如 DVCR)的数字装置以及多功能、多用途的装置(例如计算机)。

10 图 13 是图 2 中执行产生命令消息的构造部件的优选示例性结构。在图 13 中，行业标准格式 44 和消息压缩 46 的位置与在图 12 中的位置不同。为了产生完整的消息，从功能说明服务 40 选择许多文本的标准形式。以后，该消息可以由协议栈的一个较低层进行压缩。图 13 表示执行家用电器(CE)的服务或装置命令和控制的方法。消息合成可以通过 XML 标准语法定义，而压缩可以通过诸如 HTTP 那样的另一个协议层执行。命令接口语言被用于应用软件 20 的接口级，而不是更低级的硬件级。因而，该网络协议栈以表示成所述语言的命令管理，控制器 14 和被控制的装置 14 的每一个都能可以被看作用于在它们之间发送消息的、该网络的集成部分。

20 参考图 14，它显示了客户机装置 12 和服务器 14 之间交互的三个不同的实例。在第一个实例”A”中，自然人与远端服务应用程序”S”通信。该用户采用客户机装置 12 中的浏览器作为用户接口，其中浏览器控制服务应用程序”S”中的服务程序 20 并接收超文本链接标记语言(HTML)或 XML 格式的响应。还包括带有浏览器的副服务器，以便接受基于 XML 的异步命令消息发布(posting)。例如，对于 DVCR，该副服务器 14 可以接受诸如”VCR
25 FAILED: TAPE BROKE.”这样的命令消息。包括浏览器的软件代理可以被用来在浏览器的 GUI 中为用户显示命令消息，以便以后用于用户和 DVCR 的控制。基于 XML 的客户机装置 12 最好包括 HTTP 1.1 服务器能力，以便响应在其它地方启动的关于服务器到服务器的命令和控制的命令。

30 在第二实例”B”中，该用户由软件客户机控制程序 50 取代。该软件客户机控制程序 50 产生基于 XML 的命令发布给服务应用程序”S”，并接收后面的 XML 命令发布。在第三实例”C”中，该软件客户机控制程序 50



由服务器控制程序 20 这样的应用程序取代，其中命令和响应在两个服务应用程序 20 之间交换。在那种情况下，实例“B”是实例“C”在无服务时的特例。

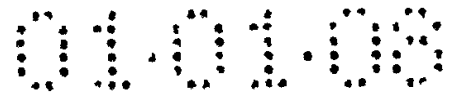
5 基于 XML 的应用程序接口语言用于第一服务器 14 和第二服务器 14(装置-装置或服务-服务)之间的控制，用于启动的万维网(Web)和启动的互联网。该应用程序接口语言基于 Web 标准，是中间设备层。在一个实施例中，装置-装置控制包括从网络 10 的另一个服务器 14 远程地控制位于一个服务器 14 中的控制程序 20 或应用程序。因而，面向这样的应用程序 20 的接口(API)利用 API 扩展在网络上进行应用。该 API 扩展最好利用诸如基于 XML 接口
10 这样的标准格式提供全面协作。

现在参考图 15，它显示了关于在网络 10 上通信的被标示为服务 A 的第一应用程序 A 以及被标示为服务 B 的第二应用程序 B 的 API 扩展的框图定义。例如，服务 A 可以是用于该网络上的第一服务器 A 的控制程序，而服务 B 可以是用于该网络上的第二服务器 B 的控制程序。服务器 B 发送命令
15 到服务器 A。对于本示例，第一和第二服务器 A 和 B 可以包括 CE 装置。

参考服务 A 的 API 扩展，第一个最上面的块 52 中提供 CE 对象的综合定义或数据库，以及用英文描述 CE 装置。该综合定义或数据库也可以以能够表示对象及其特有的方法的 C、XML 或其他格式出现。采用 XML 的该综合定义或数据库被称为 XCE 定义。对所有装置 14，第二块 54 提供用于
20 表示以 XML 格式出现的 API 的格式，被标示为接口数据类型定义 INTERFACE.DTD。

标示为工具 A 的软件代理利用关于服务 A 的 XCE 定义的一个子集，并使用关于服务 A 的接口数据类型 INTERFACE.DTD 来创建一个 XML 格式文档 NTERFACE-A.XML。该文档 INTERFACE-A.XML 根据关于服务 A 的接口数据类型 INTERFACE.DTD，描述了该服务 A 所支持的对象和方法。其
25 它的数据类型定义也可以用于创建 INTERFACE-A.XML 文档。

软件工具 A 也创建一个查找表 56，以便将来自该网络接口上的服务 B 中的消息转换成关于服务 A 的 API 定义，它以例如 C 编程并且汇编成可执行的二进制。该查找表 56 最好在汇编时创建，因此来自服务 B 的输入 XML
30 格式方法消息(命令)在运行时间期间被转换成由关于服务 A 的、被汇编的应用程序的 C 代码创建的 API 格式。该查找表 56 提供将来自服务 B 的 XML



对象方法调用，运行翻译成服务 A 的装置的本机语言调用。该查找表 56 采用装置控制程序 20 汇编，用于在服务 A 的服务器 A 上进行本地执行。

5 如果服务 A 在接收的消息中遇到错误，则 INTERFACE-A.XML 可以由服务 A 用来进行有效性检查。INTERFACE-A.XML 也可以由诸如服务 B 这样的外部应用程序用来在与服务 A 通信之前判定服务 A 的消息格式。此外，如果从服务 B 到服务 A 的消息引起错误，则服务 B 可以存取 INTERFACE-A.XML 文档来诊断该错误。

10 参考关于服务 B 的 API 扩展，第一块 58 提供诸如关于上述服务 A 的 XCE 定义的 CE 对象的综合定义或和数据库。下一块 60 用于向诸如关于服务 A 的 API 这样的远端服务或装置提供进行 XML 格式方法(命令)调用的语言定义。该语言定义是描述与该网络上的对象交互的文档型定义方法请求 CALL.DTD。

15 标示为工具 B 的软件代理至少利用关于服务 B 的 XCE 定义中的对象和方法的一个子集以及 CALL.DTD，产生用于将来自关于服务 B 汇编的 C 程序代码的命令转换为 XML 格式方法请求的查找表 62。因而，该查找表 62 例如提供在被服务 B(例如“PLAY”)调用的方法与 XML 文档或带该方法调用通过网络接口到达服务 A 的消息之间的转换。软件工具 B 使用的 XCE 定义子集依赖于该网络的使用范围和性质。例如，该子集可以选择成全部或限制使用一个家庭网络上的所有可用服务。

20 因此，该 API 扩展使用 XML 为该网络上的各装置之间提供通信。在以上的示例中，服务 B 的程序代码 20 产生对 API 的方法调用，并且该 API 调用被转换成符合关于互连装置通信的 Web/互联网标准的 XML 的 XML 格式。该 XML 方法调用(消息)被发送到该网络上的服务 A，而服务 A 重新将来自网络接口的 XML 方法调用转换成关于服务 A 的程序代码 API 定义。这
25 种转换和重新转换为网络中的不同装置提供与总是在不同装置之间要求其它二进制兼容的程序代码 API 的 Web/互联网兼容。下面显示了利用图 15 中的框图的 XML 接口块的示例。

Interface.dtd

30 写成 xml 语言、用于描述对象接口的规则

```
<!ELEMENT paramater #PCDATA>
```

```
<!ATTLIST paramater
```

```
    Type          CDATA #REQUIRED
```

```
>
```

```
5 <!ELEMENT method (#PCDATA,(parameter)+)>
```

```
<!ELEMENT method (#PCDATA,(method)+)>
```

```
interface .h
```

```
    写成 c 语言形式的示例性对象接口
```

```
10
```

```
    /* 对象 */
```

```
    typedef struct Stream {
```

```
        int id;
```

```
    };
```

```
15 /* 方法 */
```

```
    void streamPlay(int id,int speed);
```

```
    void stresmStop (int id);
```

```
interface.xml
```

```
20
```

```
    使用 interface .dtd 的规则写成 xml 语言的同一对象
```

```
<object>Stream
```

```
    <method>Play
```

```
        <parameter type= "int" >id</parameter>
```

```
25
```

```
        <parameter type= "int" >speed</parameter>
```

```
    </method>
```

```
    <method>Stop
```

```
        <parameter type= "int" >id</parameter>
```

```
    </method>
```

```
30
```

```
</object>
```

call.dtd

写成 xml 的、用于 c 函数调用的规则

```

5  <!ELEMENT parameter      #PCDATA>
    <!ATTLIST parameter
        value      CDATA #REQUIRED
    >
    <!ELEMENT method        #PCDATA,(parameter)+>
    <!ELEMENT object        (#PCDATA)>
10 <!ELEMENT call           (object,method)>

```

controller.c

写成 c 语言的示例性控制器命令

```

15  .
    .
    .
    streamPlay(0x01ae,500);
    .
20  .
    .

```

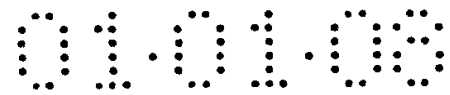
call.xml

使用 call.dtd 的写成 xml 语言的同时命令

```

25  <!--播放流的示例 -->
    <call>
        <object>Stream</object>
        <method>Play</method>
30  parameter value= "500" >speed</parameter>
    </call >

```

此外，以上提供了用于创建可用服务的描述文档的接口定义 INTERFACE.DTD 和 CALL.DTD 的示例，即如上所述的 INTERFACE.XML。CALL.DTD 定义包括产生诸如 XML 远程程序调用(RPC)或 XMLRPC 消息之类的方法调用和函数调用的规则集。CALL.DTD 定义描述了控制器服务 14 5 的输出接口。例如，在家庭网络中，INTERFACE.XML 代表在该家庭网络上可用的服务。该可用的服务是 CE 空间中整个服务的一个子集。

在一次触发记录(One-Touch-Record, OTR)情况下，用户处于诸如卫星 STB 这样的调谐接入装置的控制下。用户利用诸如表示节目清单的图形用户界面的电子节目指南(EPG)来控制调谐。OTR 记录给用户包括从 EPG 选择以后的节目用于记录的服务，而不需要用户接入 VCR 图形用户界面来为 VCR 安排一个延迟记录时间。OTR 自动控制 VCR。以下是 OTR.XML 中的动作的示例性控制清单：(1)StreamOpen=播放从卫星 STB 输出到该网络的被选中的节目流；对于 OTR，该控制是对 STB 装置的本地控制；(2)StorageOpen=开启一个存储服务；以及(3)StorageRecord=通过该网络发送一个记录命令给 15 VCR。

call .dtd

用于以 xml 语言描述 c 语言函数的规则

20 <!ELEMENT parameter #PCDATA>
 <!ATTLIST parameter
 value CDATA#REQUIRED
 >
 <!ELEMENT method (#PCDATA,(parameter)+)>
25 <!ELEMENT object (#PCDATA)>
 <!ELEMENT call (object,method)>

interface.dtd

用于以 xml 语言描述对象接口的示例

30 <!ELEMENT parameter #PCDATA>

```

<!ATTLIST  parameter
           value      CDATA #REQUIRED
           >
<!ELEMENT method      (#PCDATA,(parameter)+)>
5 <!ELEMENT object      (#PCDATA,method+)>

```

interface.xml

该文档描述提供的各 CE 服务—整个 CE 空间的一个子集

```

10 <?xml  version= "1.0" ?>
    <!DOCTYPE interface SYSTEM "interface.dtd" >
    <object>Stream
        <method>Open
            <parameter type= "int" >id</parameter>
15     <parameter type= "int" >channel</parameter>
        </method>
        <method>Close
            <parameter  type= "int" >id</parameter>
        </method>
20 </object>

    <object>Control
        <method>Set
            <parameter type= "int" >id</paxametex>
25     <parameter type= "int" >level</parameter>
        </method>
    </object>

    <object>Storage
30     <method>Open
        <parameter type= "int" >id</parameter>

```

```

        <parameter type= "int" >cannel</parameter>
</method>
<method>Record
        <parameter type= "int" >id</parameter>
5 </method>
<method>Play
        <parameter type= "int" >id</parameter>
        <parameter type= "int" >speed</parameter>
</method>
10 <method>Stop
        <parameter type= "int" >id</parameter>
</method>
<method>Close
        <parameter type= "int" >id</parameter>
15 </method>
</object>

<object>Display
        <method>Open
20 <parameter type= "int" >id</parameter>
        <parameter type= "int" >channel</parameter>
</method>
        <method>Render
        <parameter type= "int" >id</parameter>
25 </method>
        <method>Blank
        <parameter type= "int" >id</parameter>
</method>
        <method>Control
30 <parameter type= "int" >id</parameter>
        <parameter type= "int" >cid</parameter>

```

```

        <parameter type= "int" >level</parameter>
    </method>
    <method>Close
        <parameter type= "int" >id</parameter>
5    </method>
</object>

```

otr.xml

一次触发记录的 xml 语言表示

10 c 语言表示：

```

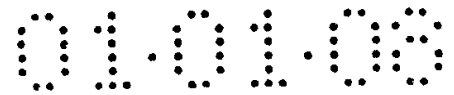
streamOpen(100,2); /* 播放一个流(由卫星馈送推送)*/
StorageOpen(24,2); /* 开启一个存储服务 */
StorageRecord(24); /* 记录该流 */

```

```

15 <?xml version= "1.0" ?>
    <!DOCTYPE interface SYSTEM "call.dtd" >
    <call>
        <object>Stream</object>
            <method>Open</method>
20             <parameter value=100" >id</parameter>
                <parameter value= "2" >channel</parameter>
        </call>
        <call>
            <object>Storage</object>
25             <method>Open</method>
                <parameter value= "100" >id</parameter>
                <parameter value= "2" >channel</parameter>
        </call>
        <call>
30             <object>Storage</object>
                <method>Record</method>

```



<parameter value= "100" >id</parameter>

</call>

象结合图 15 的上述讨论那样，第一装置 B 可以接入第二装置 A 的 INTERFACE.XML 文档，以便检查第二装置 A 的装置能力和 API 接口细节，并且判定第二装置 A 所支持的函数和命令细节。特别地，该第一装置 B 可以判定重叠，因而可用的由第一装置 B 和第二装置 A 支持的方法。图 16 是一个示例，其中包括应用程序 B 的第一服务器 B 存取包括应用程序 A 的第二服务器 A 的 INTERFACE-A.XML 文档。该第一服务器 B 包括 INTERFACE-B.XML 文档，用于与第二服务器 A 中的 INTERFACE-A.XML 文档比较。

在一个情况中，第一服务器 B 想要控制该网络中的第二服务器 A。该第二服务器 A 的 INTERFACE-A.XML 文档从第二服务器 A 传输到第一服务器 B，并被应用程序 B 用于查询该第二服务器 A 的能力和 API 接口方法。这允许第一服务器 B 利用 XML 远程程序调用 XMLRPC 来控制第二服务器 A。在另一情况中，第一服务器 B 在尝试与第二服务器 A 进行至少一次通信，但不能建立通信后，执行以上步骤。在又一情况中，第一服务器 B 还远程地查询第二服务器 A 中的 INTERFACE-A.XML 文档，而不用将该 INTERFACE-A.XML 文档传输到第一服务器 B。

在检查了 INTERFACE-A.XML 文档的内容之后，第一服务器 B 可以以上述的 XML 格式创建用于发送到第二服务器 A 的命令。一般地讲，该第一服务器 B 可以解释 INTERFACE-A.XML 文档的内容的至少一部分，它与由上述第一和第二服务器 B 和 A 使用的 XCE 定义的一个子集重叠。如果第一服务器 B 不能解释 INTERFACE-A.XML 文档的内容的某部分，那么第一服务器 B 可以通过下述的翻译忽略该部分，或者获取应用程序来帮助它解释该部分。

参考图 17，它显示了控制服务器 14 和被控制的服务器 14 之间的另一个示例性装置-装置或互连装置控制。该控制服务器 14 包括一个控制器应用程序 E，而该被控制的服务器 14 包括一个可执行应用程序 C。该被控制的服务器 14 还包括 INTERFACE-A.XML、该应用程序 C 的应用程序接口描述 A。应用程序 E 存取该被控制的服务器 14 中的应用程序接口描述 A，以便查询该被控制的服务器 14 的能力和 API 接口方法。然后应用程序 E 利用 XML



远程程序调用来命令和控制应用程序 C，以控制该被控制的服务器 14 的硬件或服务 D。日程安排(scheduler)装置可以是由诸如 VCR 中的时延记录(Time-Delay-Record)控制器的日历(time of day)驱动的控制服务器 14 的一种情况。

5 在第一示例中，应用程序 E 利用在网上的远程查询来存取应用程序接口描述 A。在第二示例中，应用程序 E 通过将应用程序接口描述 A 的一个副本从被控制的服务器 14 传输到控制服务器 14 来存取应用程序接口描述 A。然后应用程序 E 本地地查询该接口描述 A。在第三示例中，应用程序接口描述 A 被传输到为接口描述提供库空间的库装置 64，然后应用程序 E 远程地
10 查询在该库中的接口描述 A。该库装置 64 存储可用于直接控制动作和响应的有关应用程序的地址(URI)。

参考图 18, XML 协议在该网络中的各装置 14 的应用程序 20 之间的 API 级中的通信栈 66 里，提供 Web 标准的通用中间件层。在每个装置 14 中，位于该通信栈的顶上的应用程序发送和接收该网上的通信消息，并且与该装
15 置栈中，本地地控制该装置的装置硬件或服务软件的服务层通信。

标示为 XML 层输出 68 的第一 XML 层 API 用于发送消息，而标示为 XML 层输入 70 的第二 XML 层 API 用于接收消息。方法调用的 XCE 定义和 XML 定义，即，上述的文档型定义 CALL.DTD 被用于创建 XML 层输出 68。此外关于方法调用的 XCE 定义和 XML 定义，即，上述的文档型定义
20 INTERFACE.DTD 被用于创建 XML 层输入 70。例如，控制器应用程序利用 XML 层输出 68，而被控制的应用程序利用 XML 层输入 70。

参考图 19，它显示了服务器-服务器命令和控制体系结构的另一个实施例。基于 XML 的控制体系结构用于关于 Web 和互联网启动的装置或服务的装置-装置(服务对服务)控制。第一装置 A 可以利用 XML 命令消息，远程地
25 控制该网上的第二装置 B 中的应用程序 20。与每个装置的接口包括与该装置中的应用程序的接口，该接口以 XML 格式描述。所述接口可以扩展，并且可在中间件层上使用，以由该网上的其它装置进行检索和翻译，如下所述。

服务器 A 和 B 中的每个都包括用于控制该网上的其它服务器，以及由该网上的其它服务器控制的硬件和软件。在图 19 种，家庭网络装置 A 是一个
30 控制器或模块，而家庭网络装置 B 是一个被控制的装置或模块。装置 A 和 B 中的每个都包括组成接口文档 INTERFACE.XML 和文档型定义

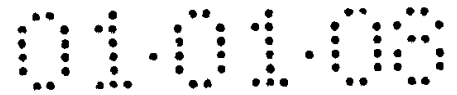


INTERFACE.DTD 的本地装置 XML 接口 72。该 INTERFACE.XML 文档包括被相应的装置 14 支持的对象描述、方法和参数。该 INTERFACE.DTD 文档可以专用于该装置的 XML 接口的有效性检查。

5 装置 A 和 B 中的每个都还包括组成用于分析和确认诸如 XML 接口和 XMLPRC 命令那样的 XML 消息的程序代码的 XML 句型分析程序 74。该 XML 句型分析程序 74 类似于在以上根据图 18 描述的所述 XML 层输入 70。此外装置 A 和 B 中的每个都还包括一个 XMLPRC 编码器和解码器(编解码器)76, 用于编码 XMLPRC 消息中的出网调用(outgoing call)的方法名称和参数, 并在它被分析之后解码引入的 XMLPRC 消息, 以便检索其中的方法名称和参数。XMLPRC 编解码器 76 独立于装置 XML 接口 72 和装置-装置控制体系结构, 因此, 允许使用不同的 XMLPRC 格式, 而不用对该装置-装置控制体系结构的任何方面作改动。

15 包括程序代码的接口获取器被装置 A 和 B 中的每个用来直接从另一个装置或从家庭网络接口库 80 获取另一装置的装置接口。当装置 14 时控制器时, 在控制器 14 中的控制器应用程序代码 82 通过监督控制器 14 中诸如 XML 句型分析程序 74、接口获取器 78 和 XMLPRC 编解码器 76 这样的软件和硬件, 实现对网上其它装置 14 的命令和控制。当装置是被控制的装置时, 位于该被控制的装置 14 中的被控制的应用程序代码 84 监督装置 14 中的软件和硬件, 以便于由其它装置 14 控制该装置 14。装置 A 和 B 的每一个中的家庭网络装置 Web 服务器 86 管理该网络上的装置之间的通信。关于装置 A 和 B 的每一个中的本机查找表 88 的 XML 由被控制的应用程序 84 用来将 XMLPRC 消息中的信息(例如方法名称、参数名称和类型)转换成装置的本机接口(例如本机方法名称、参数名称和类型)。当在 XML 消息和装置的本机接口中的方法和参数名称一致时, 不使用所述表 88。

25 装置 A 和 B 的每一个都还包括一个或多个处理器 90, 其中每个处理器 90 包括从被控制的应用程序 84 指向一个具体装置函数的本机实现的指针。在大多数装置中, 装置函数的本机实施包括运行时的二进制代码。该二进制代码在汇编时由例如包括 C 语言和 Java 等高级语言产生。因而, 家用电器制造商可以为新的功能增加更多的处理器 90, 而不会影响现有处理程序和函数的实施。装置 A 和 B 的每一个中的硬件服务 92 包括装置函数的本机实施。装置 A 和 B 的每一个都还包括含有装置函数的本机实施的 API 的本机



接口 94。

此外，诸如家庭网络对象请求中间代理(broker)(HNORB)79 和接口库(IL)80 这样的网络对象请求中间代理为家庭网络 10 提供中间件层 98。如图 19 所示，该中间件层 98 可以定位于第三装置 96 或独立的控制集线器中。

5 该 HNORB 79 包括由一装置 14 用来发现连接到网络 10 上的其它装置 14 存在的软件代理。该 HNORB 软件代理将装置名称组织成命名分层树形结构，将装置接口组织成所述可检索接口库，并将装置接口提供给请求接口信息的装置。

10 包括 HNORB 79 和 IL 80 的中间件层可以直接连接到互联网，使得被连接的家庭装置可以从本地家庭网络 10 的外部进行接入。在一个本地家庭网络中的中间件层 98 可以连接到互联网上的其它本地家庭网络中的中间件层 98，以提供包含两个家庭网络 10 的集成网络。在那种情况下，拥有适当流密码的被授权用户可以从该用户的辅助家庭网络中的电视接入该用户的基本家庭网络中的 DVD 的换片器(changer)，以便播放视频并在该电视上观看它。

15 为了使用接口库 80，在本地家庭网络上应该运行至少一个 HNORB&IL。也可使用多个 HNORB&IL。例如，一个电缆调制解调器、几个 DTV 以及一个中央家庭集线器都可以具有自己的 HNORB&IL 软件代理。为了定位 HNORB&IL，装置 14 在该本地家庭网络发送一个广播消息。响应该装置 14 的第一个 HNORB&IL 由该装置 14 利用。在 HNORB&IL 定位后，该装置 14
20 和 HNORB&IL 马上可以为登录、接口请求和提取、以及装置查找服务建立点对点传输控制协议(TCP)或用户数据报协议(UDP)连接。如果 UDP 协议不能使用，则 TCP 协议可以用于诸如 IEEE 1394 这样的高带宽连接。基于 HTTP 的 XMLPRC 也可以用于与 HNORB&IL 通信的装置。例如，装置 14 可以远程地调用 HNORB “登录”方法，以将该装置接口作为一个或多个参数传递，
25 或者，XMLPRC 调用可以从该 IL 检索部分或全部装置接口作为 XMLPRC 响应或返回值。

如前所述，一个以上的 HNORB&IL 可以在本地家庭网络 10 中同时运行，其中每个 HNORB&IL 识别可用装置的一个子集，并且一个 HNORB&IL 可以与其它的 HNORB&IL 通信，以定位它不能找到的装置 14。在一个本地
30 家庭网络 10 上的多个 HNORB&IL 可以利用诸如 TCP 或 UDP 这样的广播消息自动地相互定位。在这种情况下，多个 HNORB 构造一个分布式对象请求

中间代理，而多个接口库 80 构造分布式接口库。为了提供故障容错，如果一个 HNORB&IL 发生意外终止时，应该通知用该 HNORB&IL 登录的全部装置，并且所述装置可以自动地用其它可用的 HNORB&IL 登录。

5 每个装置接口具有相关联的一致的、唯一的逻辑名称。甚至在所述装置的位置或实际网络地址发生变化之后，其它装置可以使用所述一致的、唯一的逻辑名称来识别和接入一个装置。该逻辑名称和实际装置地址的映射由用于命名 HNORB 中的装置的软件代理管理。最好，利用标准命名方法。最好使用分层命名结构将装置名称组织成分层树。该分层结构可以使用“/”表达，它类似于文件系统中的同一符号。该结构可以利用不同方法产生，例如
10 由象家/MPEG2/TV 这样的不同服务类型产生；或由不同的位置产生，例如家/起居室/VCR。为了性能和效率的目的，几个命名树可以共存。

在图 19 中的控制服务器 A 和控制服务器 B 之间的示范型命令和控制中，中间件层 98 位于第三装置 96 中，或者可以位于独立的中央集线器中。变灰的框显示用于描述在图 19 中的特定命令和控制处理的装置元件。在一个示
15 例性操作情况中，当装置 A 和装置 B 在网络上成为可用和可接入之后，每个装置将它自己及其 XML 接口登录和提交给中央 HNORB 和 IL 中间件层 98。如果中央 HNORB 和 IL 中间件层 98 不可用，则每个装置在本地家庭网络上广播一条消息通告自己。

装置 A 的控制器应用程序 82 尝试查询被控制的装置 B 的装置接口的全
20 部或部分。如果接口库 80 不可用，则首先通过发送一个请求给网上的装置 B，然后从装置 B 接收装置 B 的 XML 接口，控制器 A 可以直接从控制器 B 请求和提取被控制的装置 B 的装置接口。但如果接口库 80 可用，则控制器 A 可以从接口库 80 请求被控制的装置 B 的装置接口的全部或部分。HNORB 的软件代理从接口库 80 结构获得装置 B 的 XML 装置接口，并将其发送回
25 控制器 A。

一旦控制器 A 接收到装置 B 的 XML 装置接口，装置 A 的控制器应用程序就使用装置 A 的 XML 句型分析程序 74 来分析和解释装置 B 的装置接口。然后装置 A 的 XMLPRC 编解码器 76 使用该句型分析程序的结果产生所需的 XMLPRC 命令消息。该 XMLPRC 命令消息被发送到该网上的被控
30 制装置 B。在接收到所述 XMLPRC 命令消息之后，装置 B 的被控制的应用程序 84 就使用装置 B 的 XML 句型分析程序 74 来分析和解释接收到的



XMLPRC 命令消息。然后装置 B 的 XMLPRC 编解码器 76 解码该句型分析程序的结果，以获得该命令消息中的方法调用信息，该信息包括执行被请求的服务的装置 B 函数的方法名称和参数。

5 然后，装置 B 的被控制的应用程序 84 使用装置 B 中针对本机查找表 88 和处理器 90 的 XML，通过装置 B 的本机接口，存取和开始装置 B 的本机函数实施。如果函数产生任何响应或返回值，则所述任何响应或返回值被编码成 XML 或 XMLPRC 消息，并且被发送到控制器 A。此外，中间件层 HNORB 和 IL 可以给控制器 A 提供被控制的装置的一个基准，因此，该装置 A 可以产生对装置 B 本机函数的远程调用，作为本地装置 A 本机函数的调用。

10 最好采用标准的 XMLPRC 格式，以便所有的装置都可以解释，和解码该网络上的 RPC 调用。因为被控制的装置 14 的装置接口可以由控制器 14 查询和检查，所以为了提高效率的目的，最好采用含有足够多装置接口信息的简化的 XMLPRC 格式。以下示例显示了关于一次触发记录(OTR)和延迟记录(TDR)操作的 XMLPRC 调用的两种可能格式。

15 示例 I:

XML RPC 调用，包括详细标记和接口信息示例性格式：

1. OTR 调用的示例：

```
<? xml version = "1.0" ? >
```

```
<call>
```

20 <object>DVCR1.record</object>

```
          <method>timeDelayedRecod</method>
```

```
          <parameters>
```

```
          <parameter>
```

```
                  <name>channel</name>
```

25 <value><int>4</int></value>

```
          </parameter>
```

```
          <parameter>
```

```
                  <name>recordTime</name>
```

```
                  <value><time>2 : 10 : 30</time></value>
```

30 </parameter>

```
          </parameters>
```

```
</call>
```

2. TDR 调用的示例

```
<xml version = "1.0" ? >
```

```
<call>
```

5

```
<object>DVCR1.record</object>
```

```
<method>oneTouchRecord</method>
```

```
<parameter>
```

```
<parameter>
```

```
<name>channel</name>
```

10

```
<value><channelName>NBC</channelName></value>
```

```
</parameter>
```

```
<parameter>
```

```
<name>startTime</name>
```

```
<value><datetime.iso8601>19990401T19:05:35</
```

15

```
datetime.iso8601></value>
```

```
</parameter>
```

```
<parameter>
```

```
<name>recordTime</name>
```

```
<value><time>2:00:00</time></value>
```

20

```
</parameter>
```

```
</parameter>
```

```
</call>
```

示例 II:

XML RPC 调用，带有精简标记和接口信息的示例性格式：

25

1.OTR 调用的示例：

```
<? xml version = "1.0" ? >
```

```
<call>
```

```
<object>DVCR1.record</object>
```

```
<method>timeDelayedRecod</method>
```

30

```
<parameter value = "4" >channel</parameter>
```

```
<parameter value= "2:10:30" >recordTime</parameter>
```

</call>

2. TDR 调用的示例

<xml version = "1.0" ? >

```

5      <call>
          <object>DVCR1.record</object>
          <method>oneTouchRecord</method>
              <parameter value=" NBC" >channel</parameter>
              <parameter
10      value = "19990401T19:05:35" >startTime</parameter>
              <parameter value = "2:00:00" >recordTime</parameter>
          </call>

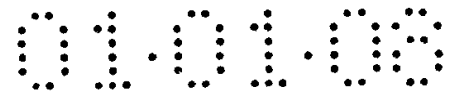
```

参考图 20，家庭装置 14 的装置接口基于采用标准化词汇的行业标准的结构化数据库 100。关于新接口和词汇的接口数据可以增加到数据库 100 种。使用英文描述全部 CE 装置的 CE 对象、方法和参数的综合定义或数据库被称为 CE 数据库 102。该综合定义或数据库可以写成 C 语言、XML 语言或其它能够表示对象及其相应的方法和参数的格式。采用标准的 XML 词汇的该综合定义或数据库称为 XCE 定义或数据库 104。

控制器和被控制的应用程序 82 和 84 使用基于 XML 的 XCE 数据库 104 的标准接口子集编程。每个装置接口和写成 XML 语言的所述应用程序 82 和 84 一起存储。虽然 XCE 数据库 104 不必写成 XML 语言，但在汇编时产生的所述子集在本发明的一个实施例中是 XML 语言的，正如以上根据图 15 所描述的那样。

在图 20 中，对于嵌入式器具 14，标示为“制造商”信息的信息在制造时内置于该器具 14 中，而在该网络中被标示作“家庭网络”的信息是该器具的操作运行时间的状况的部分。标示为 1..N 的 N 个装置 14 的装置 XML 接口 72 是标准化的 XCE 数据库 104 中的数据分支。家庭网络接口库 (HNIL)106 提供连接到该家庭网络的可用装置的装置接口集合。该家庭网络接口库 106 是整个 XCE 数据库 104 的一个子集。

在图 16 中，将装置接口从装置 A 传输到装置 B，以便于装置 B 中的应用程序 B 检查该装置 A 的接口的内容。正如上述详述的那样，装置接口包

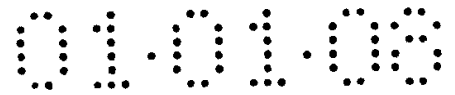


括由装置支持的对象、方法和参数的描述，例如称为装置 A 的 INTERFACE-A.XML。装置 XML 接口 72 是写成 XML 格式的装置接口。XCE 数据库 104 的内容是提供装置接口面向服务的结构。

5 参考图 20，XCE 数据库 104 还包括关于 CE 装置的一个标准化的 XCE 接口文档类型定义(DTD)，它提供一个标准化的规则集，以便使用 XML 表示 CE 装置 14。该 DTD 或其子集可以用于有效性检查。被标示为制造商工具 108 的软件代理过滤和利用关于特定 CE 装置的该标准化 XCE 定义 104 的子集，并且使用该标准化的 XCE 接口 DTD 来产生该 CE 装置的 XML 装置接口 72，例如 INTERFACE.XML 和 INTERFACE.DTD。根据该标准化的
10 XCE 接口 DTD，文档 INTERFACE.XML 包括由特定装置支持的对象、方法和参数的描述。文档 INTERFACE.DTD 是该标准化的 XCE 接口 DTD 的子集，可以用于关于该装置的 XML 接口的有效性检查。其它文档类型定义也可以用来创建 INTERFACE.XML 文档。

包括所述 XML 接口文档和所述 DTD 文档的 CE 装置的 XML 接口 72
15 存储在诸如家庭网络接口库 106 那样的通用可存取库中。软件代理收集网上的全部可接入装置 14 的装置接口 72，并将它们与该装置名称/地址信息一起放到可搜索的结构化接口库 106 中。该接口库 106 是 XCE 数据库 104 的子集，产生接口库 106 的处理类似于重建部分或全部 XCE 数据库 104 的处理。该接口库 106 可以作为家庭网络上的全部装置 14 的装置接口 72 的集合，或
20 根据存储器空间的利用率作为高速缓冲存储器，其中只有最近使用过的装置接口 72 存储在这里。在装置 14 根据诸如 DVD 播放机换盘这样的事件更新器装置接口 72 的情况下，该装置接口 72 的部分基于事件服务更新。

参考图 21，最好每个装置 14 的装置接口定义 72 均具有分层形式。这是因为对于家庭装置 14 来说，装置接口定义 72 可能过于冗长。一般来说，
25 一次存取一个或几个象用于延迟记录的单一函数这样的函数，因此，只有该装置接口 72 的一小部分被使用。仅呈现(render)装置接口 72 的一部分而不呈现整个装置接口 72 是更有效。通过使用分层的装置 XML 接口，控制器 14 可以通过在来自被控制的装置 14 或 HNORB 和 IL 中间件层 98 的、关于 XML 装置接口的请求中指定所需的函数类或多个函数组，来请求被控制的装置 14
30 的部分装置接口 72。在后一种情况下，HNORB 和 IL 中间件层 98 送回该装置接口 72 的所需部分。



参考图 21, 该分层装置接口结构可以包括四层: (1)用于每个家庭网络的 XML 接口的第一层 112, 它列出当前可用的装置, (2)用于每个装置的 XML 接口的第二层 114, 它列出函数类, (3)用于装置的每个函数类的特定的 XML 接口的第三层 116, 以及(4)用于一个函数类中的每一个函数的具体 XML 接口的第四层 118。在家庭网络里边, 只利用了三个较低级的层 114、116 和 118, 第一层 112 在该家庭网络之外利用。

图 22 是所述的层 112、114、116、118 以及相应的接口示例。每一层中的接口均经过象 XLink 或 XPointer 这样的、提供双向链接的链路链接到上层或下层(如果可用的话)。Xlink 包括含有以下两部分的超级链接功能的程序包: (1)允许同样地认出 XML 文档中的链路的 Xlink 部件, 以及(2)允许链路寻址到 XML 文档的精确子部分的 XPointer 部件。因而, Xlink 控制如何将链路插入 XML 文档, 这里的链路可以指向诸如 GIF 文档这样的数据。此外, XPointer 控制在链接到一个 XML 文档时可以从任何地方(例如从 HTML 文件)继续一个 URL 的碎片(fragment)识别符。

根据本发明, 在一个服务器 14 控制另一个服务器 14 的典型命令和控制模型中, 第一装置 14 尝试在第二接口层 114 中查询第二装置 14 的装置接口。在选择函数类(FC)之后, 第一装置 14 查询诸如记录类这样的第二装置 14 中的特殊函数类的接口层 116。此外, 该第一装置 14 可以查询诸如 OTR 或 TDR 这样的特殊函数类的接口层 118, 以便对所述函数进行调用。分层或树结构使得查找接口函数变得更有效和节约网络带宽。示例性接口文件结构和层可以是:

- 第一层 112-HN1.xml
- 第二层 114-VCR1.xml
- 第三层 116-VCR1_RecordCategory.xml
- 25 第四层 118-VCR1_RecordCategory_OTR.xml

类似地, 家庭网络接口库 106 最好是分层的, 并且以各种各样的方式结构化, 例如按装置的不同服务类型方式或按诸如房间这样的不同位置方式。所述分层结构是本地家庭网络 10 与其它家庭网络或互联网的接口。

以下显示的是可以以 XML 语法实现的示例性分层装置接口定义 72。

```
30 consumer(document_file, doc)
    +----document_file<server_home.dtd, server_auto.dtd>
```

```

+----doc(services_home,      server_auto,      server_samsung_web_site,
avc_comands, cal_commands,,)
+----services_home(xml_utility, client, server_av, lighting, comms, hvac,
utility, security, appliances, convenience,,)
5 +----xml_utility(download_DTD_file,,)
+----client(acknowledge,      attention,      error,      post_message,      sound,
stop_schedule, stop_all,,)
+----sound(alarm, ring, buzz,,)
+----server_av(controls_gen, source, sink)
10 +----controls_gen(ping, process_infor, setup,,)
+----process_info(s/wid, h/wid)
+----h/wid(ser_no, manuf, model, class,,)
+----s/wid(ser_no, exe_name, version,,)
+----setup(clock,,)
15 +----clock(hours, minutes, seconds)
+----source(service_id,      media,      rate,      protocol,      stream_format,
controls_av,,)
+----sink(service_id,media, rate, protocol, stream_format, controls_av,,)
+----service_id(url,,)
20 +----media(tpt_stream, ram, disk, tape,,)
+----disk(name, number,,)
+----rate<value>
+----protocol(61883/1394, UDP/IP/Ethernet,,)
+----61883/1394(isoch_ch_no)
25 +----stream_format(video, audio,,)
+----video(dv, mpeg2tpt, dsstpt, mpeg2pes, mpeg10801-tpt,)
+----audio(mpeg3, ac-3, midi,,)
+----controls_av(flow_control, tune, timer_record, ui_control,,)
+----timer_record(tune, flow_control)
30 +----flow_control(play, stop, goto, record,,)
+----play(time_params)

```

```

+----record(time_params)
      +----time_params(now, start, duration, end,,)
+----tune(send_epg, channel,,)
      +----channel(number, id, time_params,,)
5      +----ui_control(dispiay, acoustic)
      +----display(brightness, contrast, color/tint, horiz_size,
vert_size,,)
      +----acoustic(volumn, base, treble, balance, fade,)
+----lighting(sensors, lights, send_epg)
10      +----sensors(living_room, sky,,)
      +----lights(rooms_up, rooms_down, yard,,)
      +----rooms_up.bed1, bed2, bed3, bed4,,)
      +----bed1(lamp, dimmer,,)
      +----dimmer<value>
      +----rooms_down(family, kitchen, living, dining, soho, garage,,)
15      +----yard(front, back)
+----comms(homehub, intercoin, telco,)
+----homehub(send_device_list,send_configuration, send_snmp_mib,,)
      +----intercom()
20      +----telco()
+----hvac(controls_gen, controls_havc,,)
      +----controls_hvac(a/c, heat, temp, humidity,)
      +----temp(low, high, hysteresis,,)
+----utility(meters, energy_mgmt,,)
25      +----meters(water, gas, electric,,)
      +----water<value>,gas<value>,electric<value>
+----security(sensors, send_epg, alarm,,)
      +----sensors(peripheral, motion,,)
      +----peripheral(rooms_up, rooms_down,,)
30      +----motion(room_down, yard,,)
+----appliances(microwave, range, oven, fridge, freezer, coffee, toaster,

```



```

washer, dryer, water_heater,,)
    +----microwave(send_epg, controls,,)
    +----fridge(temp,,)
    +----water_heater(temp)
5    +----convenience (window, curtain_open, door/gate, pool/spa, bath,
    fountain, lift, jacuzzi,,)
    +----curtain_open<value>
    +----server_auto(message, server_anto_ford_explorer_98,,)
    +----server_auto_ford_explorer_98(mileage, maintenance,,)
10   +----mileage<data>
    +----maintenance<data>
    +----server_samsung_web_site(message, service, help,,,)
    +----avc_commands<,,,command_string,,,>
    +----service_id url,,)
15   +cal_commands<,,,command_string,,,>
    +----service_id url,,)

```

所述的分层装置接口定义 72 可以包括下列字段：

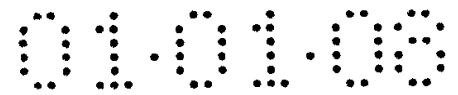
‘document file’ 名称，它提供该文档类型定义(DTD)文件的名称，该文件可以由 XML 句型分析程序 74 用于确认 XCE 数据库 104、或者 XCE 数据库 104 的 XML 版的部分的合法性和正确性。针对 XCE 结构的不同部分可以存在几种 DTD 文件，其中所述多个 DTD 不同于关于针对通信的 RPC.CALL 和 INTERFACE.DTD 文档类型定义。

‘doc’ 名称，提供覆盖能力、属性、通信和控制接口的区域的顶级名称。

25 ‘Server_home’，提供关于家庭自动化、家用电器、公用设施等的区域。

‘Server_auto’，用于车库的汽车，显示可用于一种或多种汽车类型的消息接口。例如，server_anto_ford_explorer_98 是关于一种特定汽车的接口。这允许接入该汽车的里程和维护接口，并且还可以用来例如由汽车制造商或车库远程接入，用于检修和远程诊断。

30 ‘server_samsung_web_site’，提供与家庭之外的制造商 Web 站点的通



信。包括消息、服务、帮助等接口。

5 ‘AVC_commands’和‘CAL_commands’，例如提供能够解释AV/C和CAL语言的传统(legacy)装置。该结构的这一部分识别所述语言中的命令，这里的命令以XML语言标注和携带。因而，该内容不是XCE(Web)对象，并且协议转换应用程序可以被用于与原始CAL或AV/C的应用软件的接口。

在以上描述种，‘Server_home’提供包括A/V家用电器的主结构。详细展开了该结构的关于视频服务接收信宿(sink)和流目的地(例如DVCR)控制接口的具体示例的分支。典型的家庭网络中的控制接口可以包括：

10 ‘xml_utility’提供支持诸如下载一个被更新的DTD文件、接口文件、程序文件等那样的实用网络功能的细节。

‘client’描述包括Web浏览器的客户机装置12的接口部分。例如，‘acknowledgment’指示被发出的消息和或命令的确认的控制器验收。

‘server_av’为所有可利用的音频和视频服务提供控制和负载接口，包括STB、DVCR、DTV、DVD、AUDIO等。

15 ‘lighting’提供与家庭自动化照明控制器的接口，包括传感器、灯等。

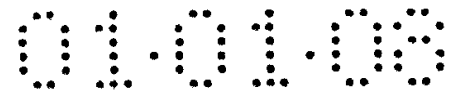
‘comms’提供与通信装置的接口，一般是为了装置设置或参数的实用目的或远程管理，或者为了恢复配置。

20 ‘hvac’为HVAC系统提供远程控制接口，可以用来从例如家庭之外的公共公司控制所述系统，以便在每天的最大负载时期内关闭家庭HVAC系统。此外，所述接口可以用来通过一种器具从家庭中控制HVAC系统，以使得基于装置的控制提供比自动调温器控制更精确的控制机构。

‘utility’，例如提供用于读取家庭供给表(utility meter)的接口。

25 ‘security’提供用于安全传感器和警报设置的接口。因而，使用该接口，运行在家庭网络装置上的应用程序可以接入位于家庭周围的传感器和检测装置，用于监视和控制那些装置。

‘appliances’提供用于厨房、公用或普通家庭用具的接口，例如包括提供远程控制或监视温度设置或其他来自控制器的控制和参数。在一种情况中，微波用具可以扫描食物包装袋上的条形码信息，并且接入制造商的数据库来获得适合已知类型的微波系统烹饪该食物的时间。使用装置对装置的命令和控制的用具的这种集成，提供很多用来提供诸如当在厨房或客厅接电话时，自动暂停洗碗机和让电视静音这样的服务的控制情况。



‘convenience’ 提供与用于提供方便服务的装置的接口，例如与窗帘、窗户、百叶窗、涡流控制器的接口。

5 在以上的描述中，‘server_av’ 是关于提供 A/V 流服务的 A/V 用具的控制接口结构的部分，并且被分成 ‘controls-gen’、‘source’ 以及 ‘sink’ 能力。

10 ‘controls-gen’ 提供用于装置制造商属性和进行诸如查验(ping)测试装置的存在普通实用程序的接口。此外，还可以包括造在其中的诸如软件和硬件标识和版本信息这样的属性。提供这种接口的装置返回提供所述软件名称或标识的数据，而不会影响任何控制作用。还可以包括设置日历钟的时间的接口。

15 ‘sink’ 提供用于介质流服务装置的接口。该结构根据所提供的服务(即视频流记录和播放)而不是根据诸如 VCR 这样的特定装置名称进行组织。例如，对于带有视频节目格式的网络来说，调谐器和 DVD 播放器都是视频节目流源，并且可以对诸如开始和结束进行控制。特定装置的控制中的差异由该定义结构的较低层寻址。

‘source’ 提供类似于 ‘sink’ 接口的接口。

20 参考以上的描述，‘service_id’ 或 ‘application_interface_id’ 包括一个或多个装置 14 的名称、地址或 Web 地址或 URL 位置。因为 XCE 数据库 104 包括全体约定接口，所以一般来说，动态主机配置协议(DHCP)软件代理给每一个装置分配一个地址和一个缺省名称，并且将该地址和名称增加到该服务或装置的接口。然后，软件代理 110 从本地地连接到该家庭网络的全部装置收集包括子集和或 ‘装置的部分 XCE’ 定义的装置接口 72，以产生 ‘网络的部分 XCE’。附加的有关外部接口可以增加到关于外部控制的结构中。例如，按照连接到该网络的装置的装置接口，‘service_id’ 可以是接收到的结构或包括软件代理中的项目的网络接口库 106 中的名称/地址。这以后用户可以在该数据库中查找服务，并且可以使用所述名称/地址存取其接口包括该库的特定数据分支的应用程序。因而，该网络可以包括多个由所述名称/地址信息区分的相同服务。

30 ‘media’ 提供关于介质类型的接口，这些介质例如包括来自调谐器的传输流、来自 PC DRAM 的 RAM、用于 CD 和 DVD 的盘以及磁带。可以命名和识别这些介质，并且控制器可以查找 XCE 数据库来识别当前设置在网

5 络中的介质。当在网上提供诸如 DVD 之类的新介质时，装置 72 中识别在盘上的节目材料的那部分因此变化。因而，不必传输整个装置接口 72 给 XCE 数据库，而仅仅传输有关的部分即可。在接收引起注意的信号时，库软件代理 110 可以获取新的更新，并且将其放在接口库 106 的适当位置。磁盘介质的增加类似于增加服务给网络，或者连接其它用具到网络。

‘rate’ 提供用于装置接口的数据流速率值，例如 6M 位/每秒或 19.2 位/每秒。

‘protocol’ 识别用于所述数据流的协议。如果提供一种以上的协议，例如 61883/1394 或 UDP/IP，则选择所要求的协议。

10 ‘stream_format’ 提供关于数字流音频和视频片断的打包格式和/或压缩标准。如果提供一种以上的格式，则经由接口消息选择所要求的格式。控制器应用程序 82 可以检查可用的格式，以判定是否存在兼容的。

‘controls_av’ 为 A/V 介质用具提供主控接口。

15 ‘Flow_control’ 提供诸如 PLAY、STOP、GOTO、RECORD 等之类的数据流控制作为用于特定装置的方法。这些方法除了例如 PC 软件之外，对嵌入式用具不发生变化。该控制可以包括用于延迟操作的时间参数。

‘Tuning’ 提供用于调谐控制的接口。控制器 14 可以发送请求给被控制的装置 14 的接口，而被控制的装置 14 送回如上所述的电子节目指南(EPG)数据结构。

20 ‘UI control’ 提供与被控制的应用程序 84 的接口，以控制关于显示的调整，例如亮度和对比度，以及关于音频的调整，例如音量和低音。

‘Timer_record’ 提供用于控制器应用程序 82 的设置数据的接口，以实施延时记录。可以利用直接频道调谐信息和流控制(time_aparams)信息。

25 以上的描述可以均等地用于客户机装置 12。可以利用关于 CE 空间的一种替换语法 XCE 定义或数据库。该替换语法 XCE 数据库包括，例如，含有家庭自动化、用具和汽车在内的全部服务描述。在服务对象提供用于控制的挠性(flexibility)和参数的情况下，利用控制方法按要求控制该对象。以下显示以 AV/C 和 CAL 命令语言写成的示例性命令，包括二进制和十六进制数据串。

30 consumer(document_file, doc)
+----document_file<server_home.dtd, server_auto.dtd>

```

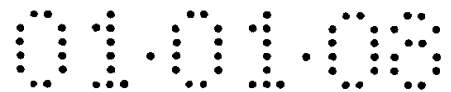
+----doc(avc_comands, cal_commands, services_home, server_auto,
server_samsung_web_site, server_anto_ford_explorer_98,,)
+----avc_commands<...command_string...>
+----cal_commands<...command_string...>
5 +----services_home(client, av, lighting, comms, hvac, utility, security,
appliance, convenience,,)
+----xml_utility(download_DTD_file,,)
+----client(acknowledge, attention, error, post_message, sound,
stop_schedule, stop_all,,)
10 +----sound(alarm, ring, buzz,,)
+----server_av(source, sink)
+----source(service_id, media, rate, protocol, stream_format,
controls_gen, controls_av,,)
+----sink
15 +----service_id(url,,)
+----media(tpt_stream, ram, disk, tape,,)
+----disk(name, number,,)
+----rate<value>
+----protocol(61883/1394, UDP/IP/Ethernet,,)
20 +----61883/1394(isoch_ch_no)
+----stream_format(video, audio,,)
+----video(dv,mpeg2tpt,dsstpt,mpeg2pes,mpeg1080i-tpt,)
+----audio(mpeg3, ac-3, midi,,)
+----controls_gen(ping, process_info, setup,,)
25 +----controls_av(flow_control, tune, timer_record, ui_control,,)
+----process_info(s/w_id, h/w_id)
+----h/w_id (ser_no, manuf, model,class,,)
+----s/w_id (ser_no, exe_name, version,,)
+----setup(clock,,)
30 +----clock(hours, minutes,seconds)
+----time_record(tune, flow_control)

```

```

+----flow_control(play, stop, goto, record,,)
      +----play(time_params)
      +----record(time_params)
+----tune(send_epg, channel,,)
5      +----channel(number, id, time_params,,)
      +----time_params(now, start, duration, end,,)
+----ui_control(dispiay, acoustic)
      +----display(brightness, contrast, color/tint, horiz_size,
10      vert_size,,)
      +----acoustic(volumn, base, treble, balance, fade,)
+----lighting(screen, light, send_epg)
+----sensors(living_room, sky,,)
+----lights(rooms_up, rooms_down, yard,,)
      +----rooms_up.bed1, bed2, bed3, bed4,,)
15      +----rooms_down(family, kitchen, living, dining, soho,
      garage,,)
      +----yard(front, back)
      +----bed1(lamp, dimmer,,)
      +----dimmer<value>
20 +----comms(netmam, intercom, telco,)
      +----netman(send_device_list,      send_configuration,
      send_snmp_mib,,)
      +----intercom()
      +----telco()
25 +----hvac(controls_gen, controls_havc,,)
      +----controls_hvac(a/c, heat, temp, humidity,)
      +----temp(low, high, hysteresis,,)
+----utility(meters, energy_mgmt,,)
      +----meters(water, gas, electric,,)
30      +----water <value>, gas<value>, electric <value>
+----security(sensors, send_epg, alarm,,)

```



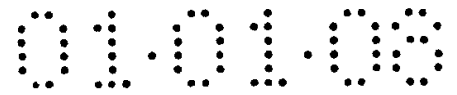
```
+----sensors(peripheral, motion,,)
      +----peripheral(rooms_up, rooms_down,,)
      +----motion(room_down, yard,,)
+----appliances(microwave, range, oven, fridge, freezer,
5 coffee, toaster, washer, dryer, water_heater,,)
      +----microwave(send_epg, controls,,)
      +----fridge(temp,,)
      +----water_heater(temp)
+----convenience (window, curtain_open, door/gate, pool/spa,
10 bath, fountain, lift,,)
      +----curtain_open<value>
+----server_auto(message, mileage, maintenance,,)
      +----mileage<data>
      +----maintenance<data>
```

15

另一方面，本发明使用对关于网络中的装置-装置命令和控制的现有的命令语言实施。装置包括内部对象和在运行时间根据现有的传送机构创建二进制串的 API。在这种情况下，为了提供从该网中的一个装置 14 对网中的其它装置 14 的 XML 远程程序调用(XML RPC)，要用对 XML 服务 API 的调用替代现有的应用程序接口实施。因而，原来的实施等效于 XML 服务 API 的包装(wrapper)。图 18 还显示了使用诸如虚线框中的 CAL 或 AV/C 之类的其它命令语言创建的应用程序，它们的接口实现用 XCE/XML 服务 API 中的包装替代。以下显示了用于将 CAL 命令语言变成 XML RPC 格式的示例。

25 现有实施：

```
-----
void DeviceCALCommand (int command) {
    .
    .
    /*
30 创建 CAL 格式化的字节串来表示这个对象/方法，并输出给线路
    */
```



```
CreatCALFormattedByteString( command);/* 对每种协议都不同*/  
SendCALByteString();/* 对每种协议都不同*/  
}
```

5 包装的 XML 服务 API 调用:

```
void DeviceCALCommand (int command)  
{  
    /*  
    用对 XML 服务 API 的调用替代 CAL 实施  
    */  
    CreateXMLMessage (command);/* 总是相同 */  
    sendXMLMessage ();/* 总是相同 */  
}
```

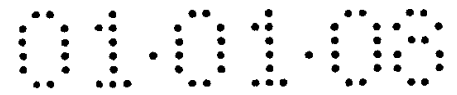
10

15 参考图 23, 另一方面, 本发明提供用在网络的不同装置之间的装置间通信的标准命令协议和控制语言翻译。对于共享信息的不同装置, 该信息必须采用请求的装置可以解释的格式。为了让装置 120 控制另一装置 22, 这两个装置都必须使用通用语言来解释相互之间的命令。本发明提供用于数据和命令协议的通用识别格式。

20 在一个实施例中, 提供数据和命令协议的通用表达或打包方法, 接收装置 122 由此可以判定被传输的数据本机格式。如果接收装置 122 能够解释本机格式, 则可以直接接收该数据。否则, 接收装置 122 可以请求翻译装置 124 或应用程序将该数据翻译成请求装置 122 可以解释的所需格式。翻译装置 124 或应用程序判定该原始数据的本机格式, 将该数据翻译成所述的所需格式, 并且将被翻译的数据发送到请求装置 122。

25 因而, 即使该数据本来是由发送装置 120 以请求装置的本机语言格式提供, 请求装置 122 也处理那个数据。请求装置 122 还以请求装置的本机格式发回一个响应给发送装置 120, 或者通过用于翻译成发送装置 120 的本机格式的翻译装置 124 或应用程序、由代理发送一个响应。该翻译方法可以用于包括命令协议、数据文件和音频/视频流的信息。

30 对于不使用上述通用格式的装置, 本发明提供将包括命令协议在内的数据翻译到这样的不一致的装置, 以及从这样的不一致的装置将包括命令协议



5 在内的数据翻译出来的服务。例如，当一个不一致的装置 120 发送数据到一致的装置 122 时，一致的装置 122 可以根据该数据的本机格式的判定来翻译该数据。例如，一致的装置 122 可以为该数据内的特定位模式而检查该数据。当一致的装置发送数据到已知的不一致的装置时，该一致的装置可以在传输之前、根据该不一致的装置的本机格式的判定翻译该数据。

一个示例性实现可以用于支持 IP 和 HTTP 协议的家庭网络。该家庭网络可以连接到互联网，来为所需功能获得各种应用程序和服务。因而，可以使得通用格式法与互联网协议和在互联网和家庭网络上操作的程序兼容。

10 提供通用数据格式一个示例是利用 XML 为在家庭能够网上传输的数据创建一个包。该数据可以包括命令协议、流式音频或视频、图形或应用程序。该数据用识别该包的数据和内容的本机格式的、写成 XML 语言的一个标准首标“包装”。该首标允许唯一标识该 XML 代码的数据部分的数据类型，因此，如果需要的话，可以翻译该数据，并且在接收后提供给适当的应用程序。

15 在 Web 标准下，该标识处理由浏览器使用识别文件传输的类型和内容的文件名扩展来执行。然后该浏览器运行适当的插件程序来处理该文件。在家庭网络中，XML 用于识别数据传输，它用上述的通用标识方法提供基于 IP 的所有家庭网络传输。

20 另外，可以在家庭网络协议栈中提供软件层，以便唯一地识别该家庭网络上的所有数据传输的内容。该软件层可以替代 XML 使用。本发明的通用格式和标识原理均等地运用于作为标识方法的使用 XML 的实施例或者所述软件层。

25 在图 23 中，在接收到数据包传输后，接收装置 122 检查该数据包的 XML 识别首标以判定其中的数据格式。如果数据格式为可由装置 122 辨认，则放弃 XML 识别首标信息，而让该装置直接处理该数据。否则，装置 122 将接收到的 XML 包转换成 XML 翻译请求包，并发送该请求包和数据给翻译服务器 124。

30 翻译服务器 124 翻译该数据，并将翻译过的数据转换成 XML 翻译响应包。然后翻译服务器 124 将该响应包转送回请求装置 122。在翻译出错的情况下，翻译服务器 124 可以将翻译响应出错情况提供给请求装置 122。在接收到被翻译过的数据后，请求装置 122 处理在该响应包中被翻译的数据。

XML 数据包或分组的示例可以是：

<IDENTITY type=format=AV/c>... 包数据 ... <IDENTITY>

翻译请求包或分组的示例可以是：

<TRANSLATION REQUEST type=Command format=CAL>

5 <IDENTITY type=Command format=AV/C> . . . packet data. . .
<IDENTITY>

<TRANSLATION REQUEST>

翻译请求包或分组的示例可以是：

<TRANSLATION RESPONSES type=Command format=CAL>. . .

10 packet data . . .

<TRANSLATION RESPONSES>

翻译响应出错情况包或分组的示例可以是：

<TRANSLATION RESPONSES type=Command format=CAL>. . .

packet data . . .

15 <ERROR condition=Unrecognized command>Translation could not be
performed<ERROR>

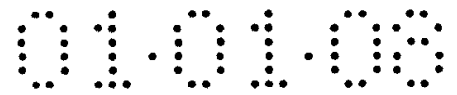
<TRANSLATION RESPONSES>

此外，图 24 中的表 3 包括包或分组类型和格式的部分清单。

20 为了提供翻译服务，以类似于 DHCP 服务器的方式，在网络配置期间，
在该网上识别翻译服务器 124。在配置了网络之后的一段连续时间中，翻译
服务器 124 向该网络中的所有装置广播其 IP 地址。当在网络启动期间，在
网络上广播时，与该翻译服务兼容的所有装置 120、122 存储翻译服务器 124
的 IP 地址。

25 另外，请求装置 122 可以在家庭网络上广播翻译请求。在网中接收到该
翻译请求的所有翻译服务器 124 通过发送一个翻译响应给请求装置 122，来
响应该翻译请求。然后请求装置 122 在响应的翻译服务器中间选择一个翻译
服务器 124。在一个示例中，请求装置 122 选择响应该翻译请求的第一翻译
服务器 124。在另一示例中，翻译服务器 124 可以在它们之间和/或与请求装
置 122 达成协议，用于选择满足该翻译请求的翻译服务器 124。

30 在本发明的一个实施例中，利用多个翻译服务器 124 来完成全部翻译请
求。例如，单一翻译服务器 124 可能没有能力翻译所有请求。在这种情况下，



需要识别每个翻译服务器 124 的地址以及每个翻译服务器 124 可以提供的翻译服务类型。每个装置 120、122 可以存储所有翻译服务器的 IP 地址的清单以及每个翻译服务器 124 提供的翻译服务的类型的相应清单，和任意相关的翻译应用程序。

5 出于效率的考虑，如果发送装置 120 希望发送数据到已经知道使用了不同于该发送装置 120 的格式的本机格式的接收装置 122，则发送装置 120 可以通过翻译服务器 124，由代理发送该数据到接收装置 122。发送装置 120 传送一个类似于翻译请求命令的命令到翻译服务器 124，并包括接收装置 122 的地址作为翻译过的数据的目的地址。

10 在接收装置 122 需要翻译数据流的情况下，发送装置 120 可以直接发送该数据流到翻译服务器 124，而翻译服务器 124 接下来传送翻译过的数据到如上所述的接收装置 122。另外，发送装置 120 可以发送该数据流到接收装置 122，然后接收装置 122 发送该数据流到翻译服务器 124，以便翻译和返回被翻译过的数据到接收装置 122。

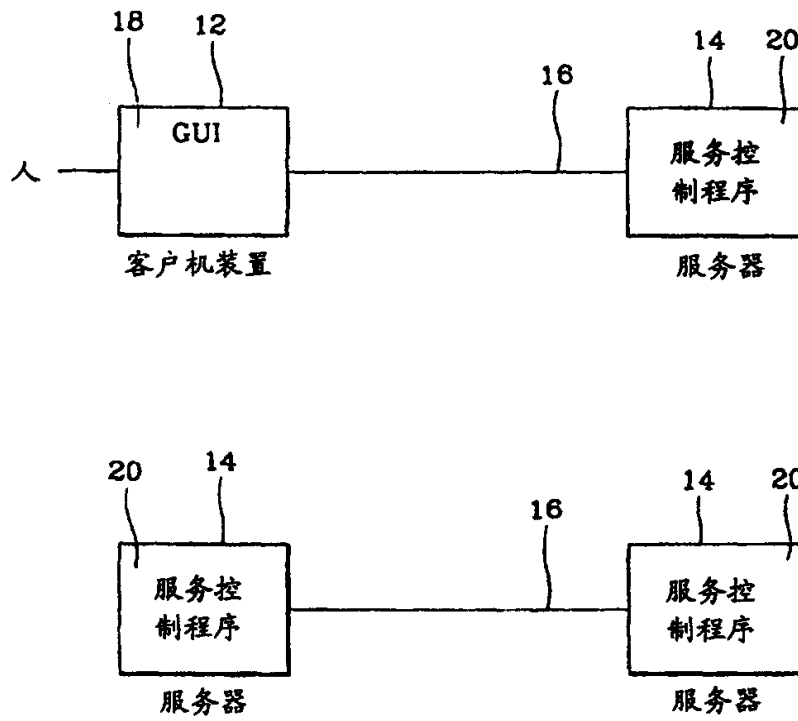
15 在这里的描述中，控制机制是基于超文本传输协议(HTTP 1.1)的，该协议提供用于分布式协作超介质信息系统的应用程序级协议。在对于多任务的广泛应用中，HTTP 充当普遍使用的、无国界的(stateless)、面向对象的协议。HTTP 的一个特点是数据表示法的键入(typing)和协商，它允许在独立于被传输的数据的情况下建立系统。由家庭网络上的装置和应用程序使用的网络协议最好是 IP(互联网协议)。但也可以使用其它协议。

20 虽然借助于其优选实施例将本发明进行了相当详细的描述，但还可能有其它的实施例。因此，附加的权利要求不应该被限制在含在这里的优选实施例的描述中。

25

说明书附图

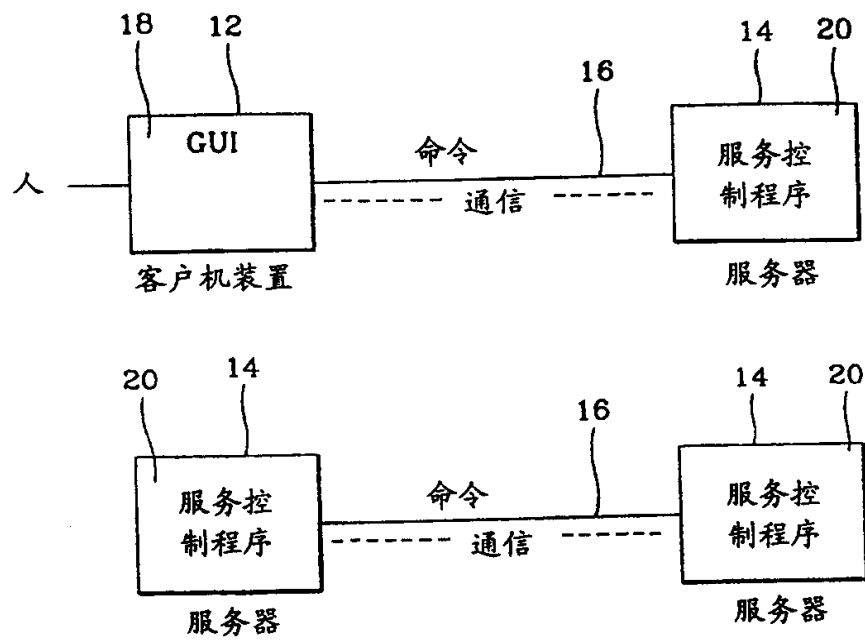
构件块或模块



10

图 1

控制系统



10

图 2

多客户机装置多服务器家庭网络控制模式

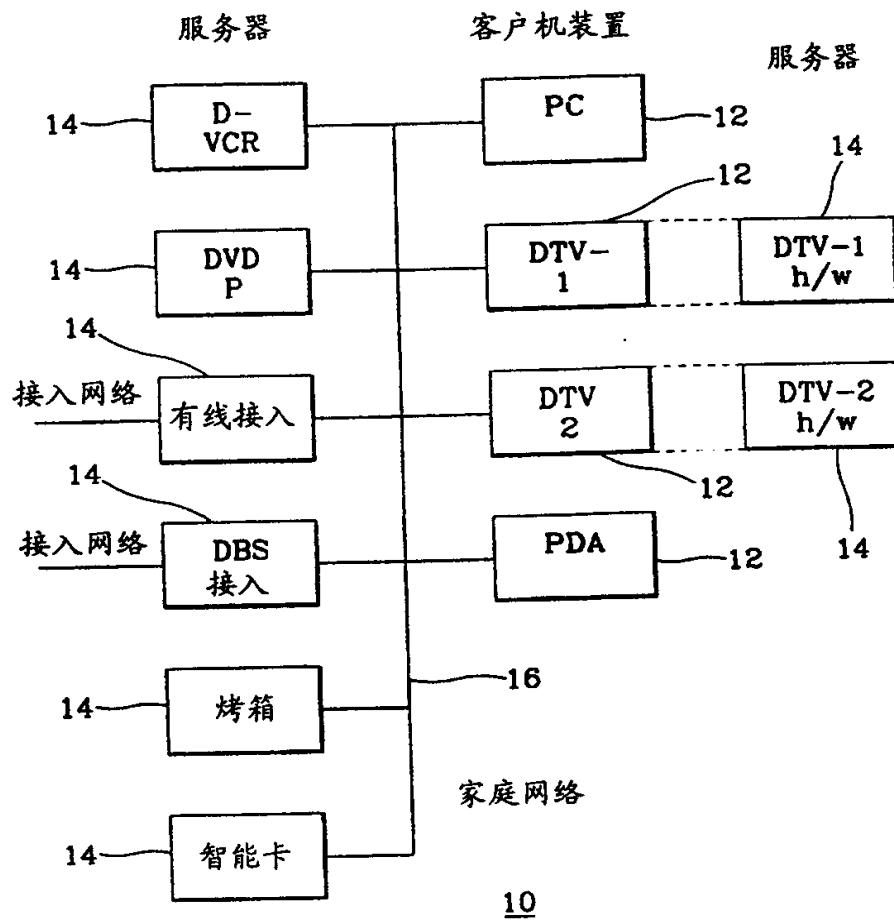


图 3

模块元件GUI控制

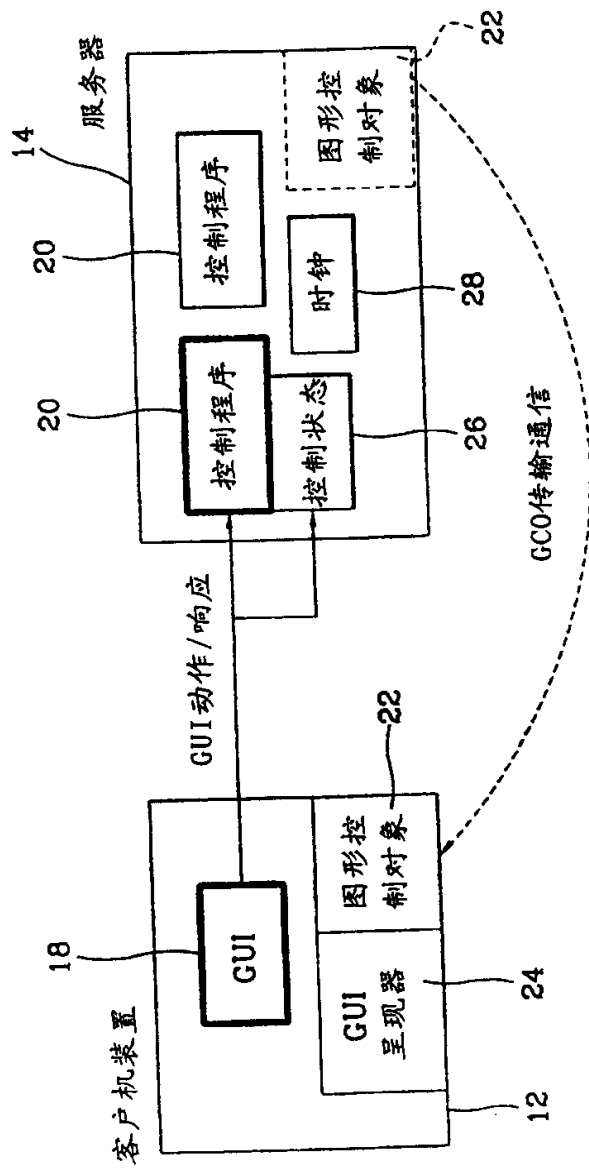


图 4

客户机装置框

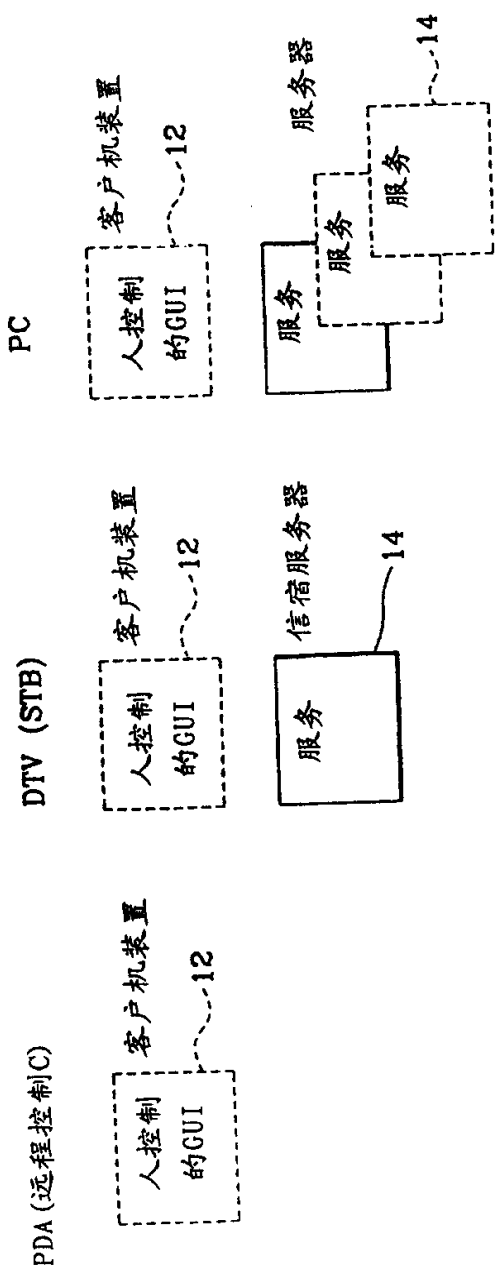


图 5

服务器/服务器CL的模块元件

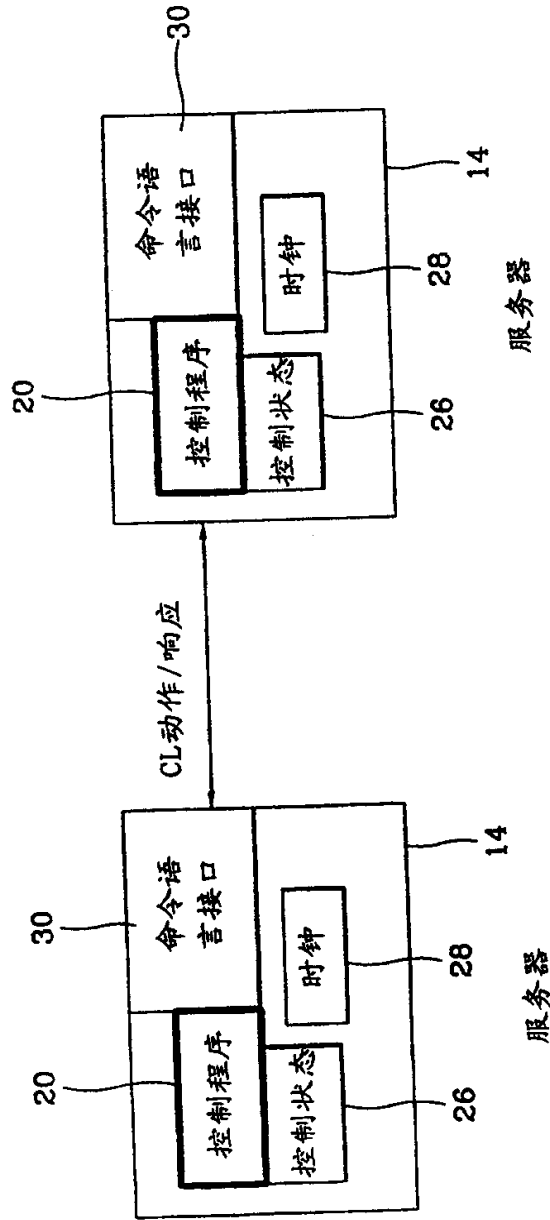


图 7

基本A/V、客户机/服务器/服务器模式

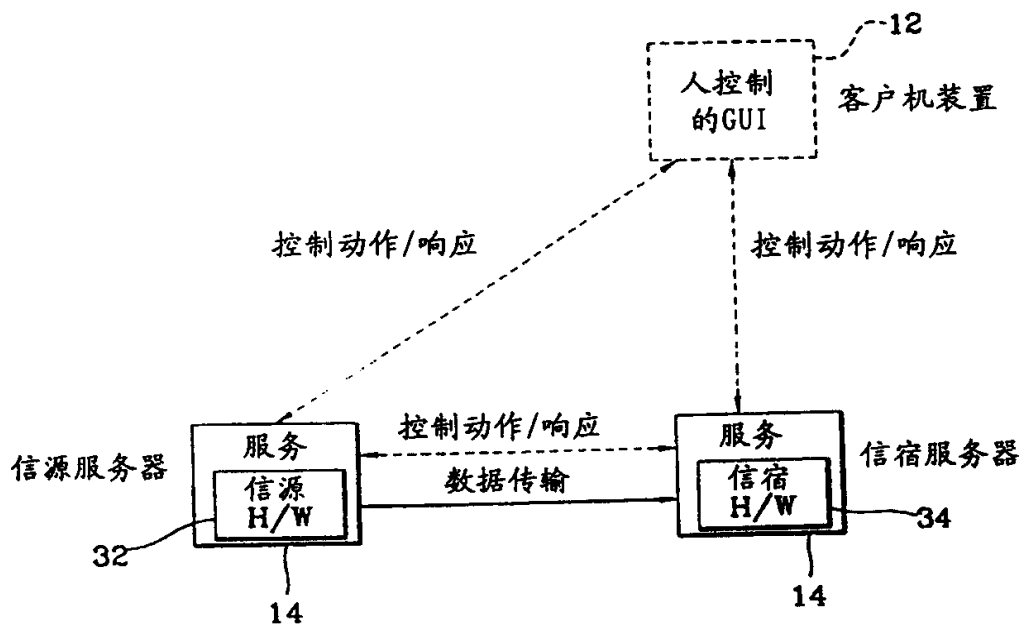


图 8

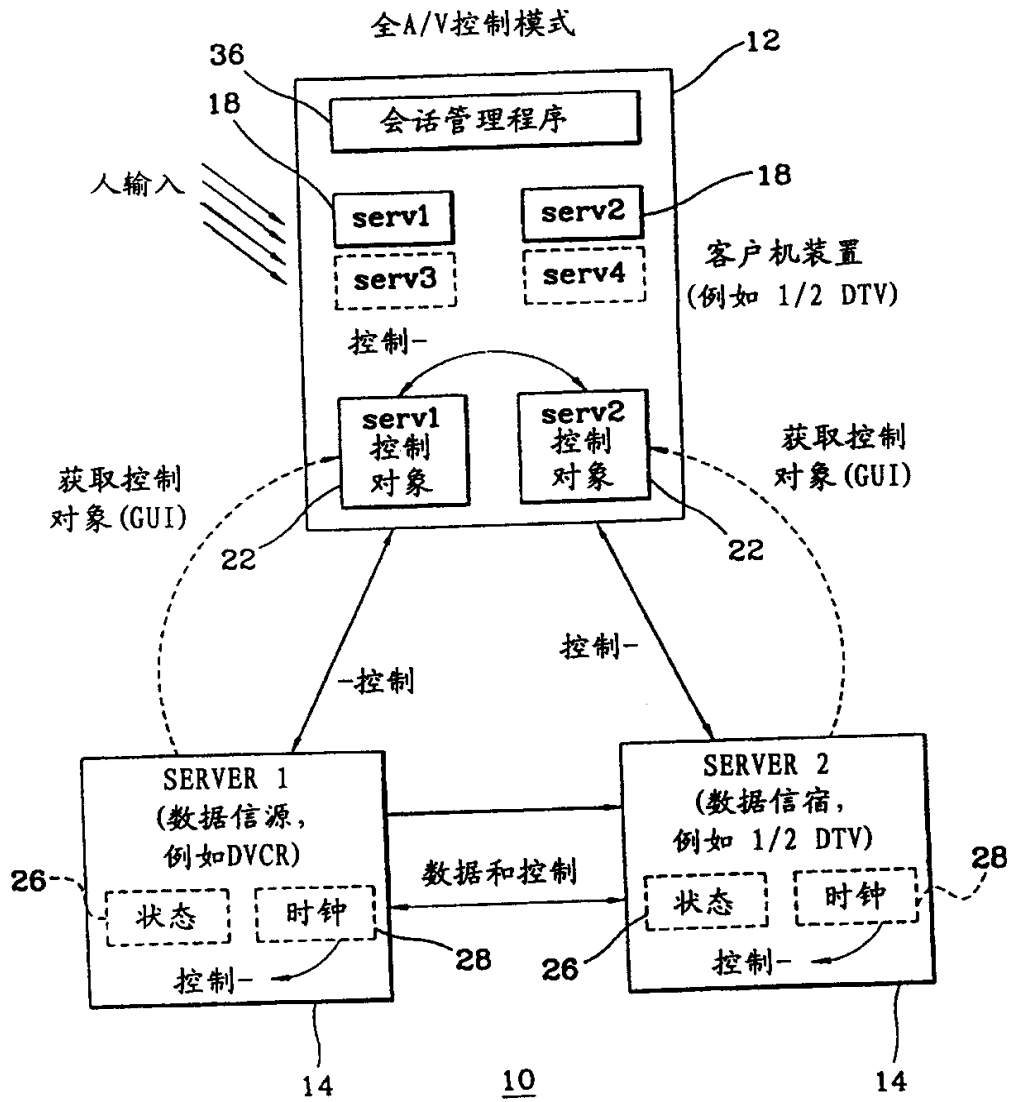


图 9

表 1: 能力表

服务	信宿/信源
服务 A	信宿
服务 B	信源
服务 C	信源
服务 D	信宿

图 10

表 2: 属性表

名称	长度	值
装置制造商	20 字符	装置制造商名称
制造商URL	60 字符	装置制造商的主页URL
制造商图标	20 字符	装置制造商的图标名称
装置名称	30 字符	装置名称
装置模式	20 字符	装置模式号
装置类型	20 字符	装置类别
装置位置	30 字符	装置组成位置
缺省信源	15 字符	数据类型、缺省信源装置(IP地址)
缺省信宿	15 字符	数据类型、缺省信宿装置(IP地址)
装置图标	20 字符	装置图标名称

图 11

0.0.0.0

由CAL和AV/C使用的消息结构

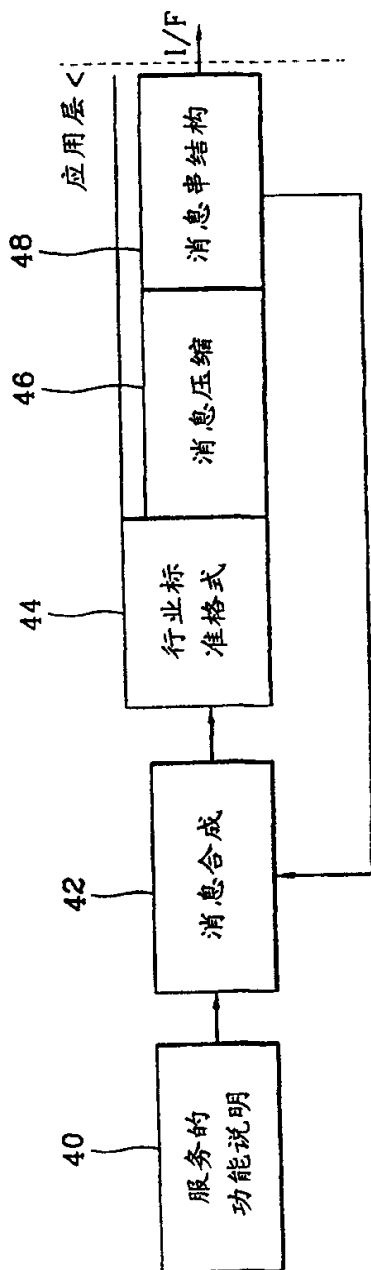


图 12

010108

新XCE/XML的消息结构

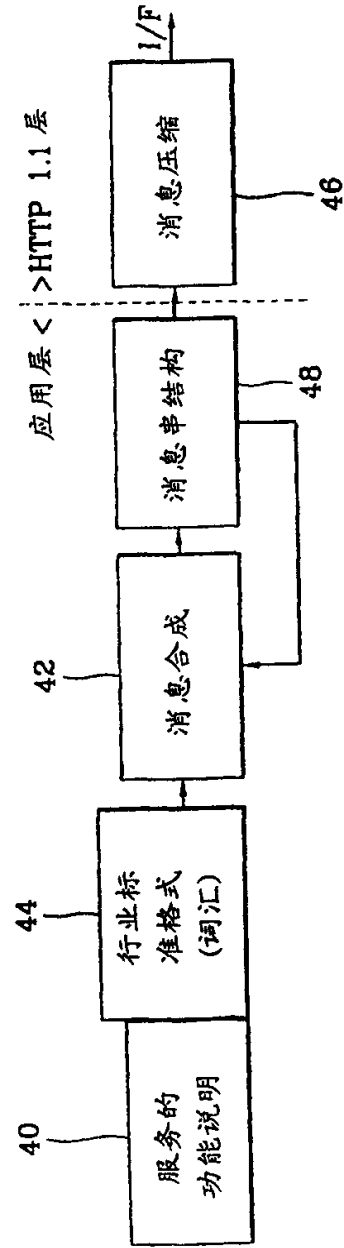


图 13

利用Web和XML的客户机-服务器模式

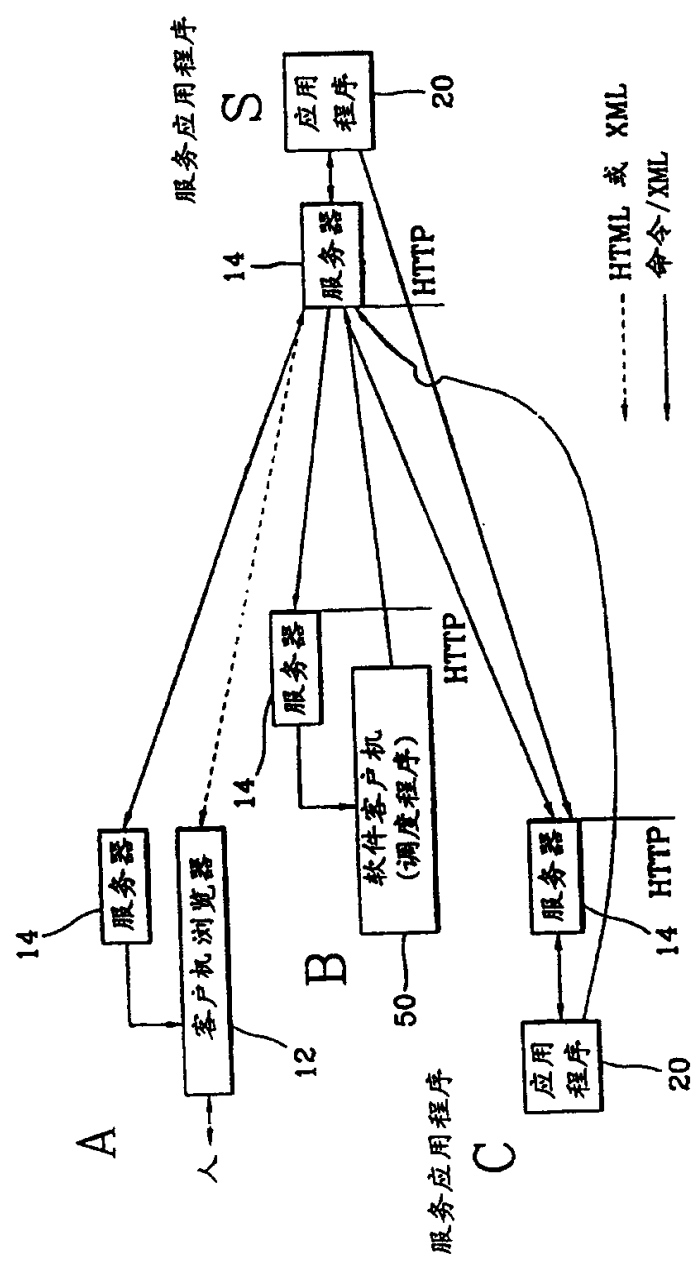


图 14

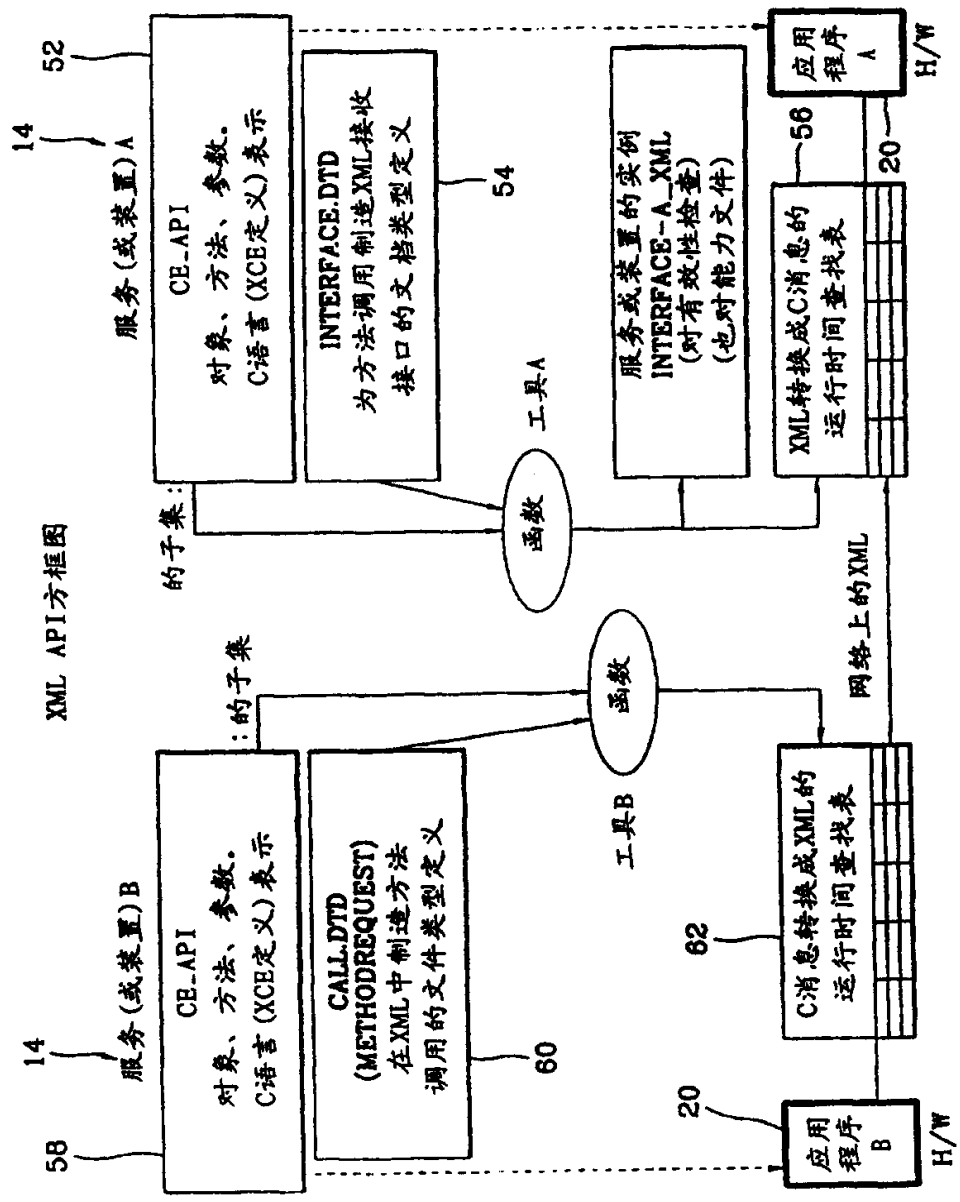


图 15

传输和学习外部INTERFACE-INSTANCE.XML

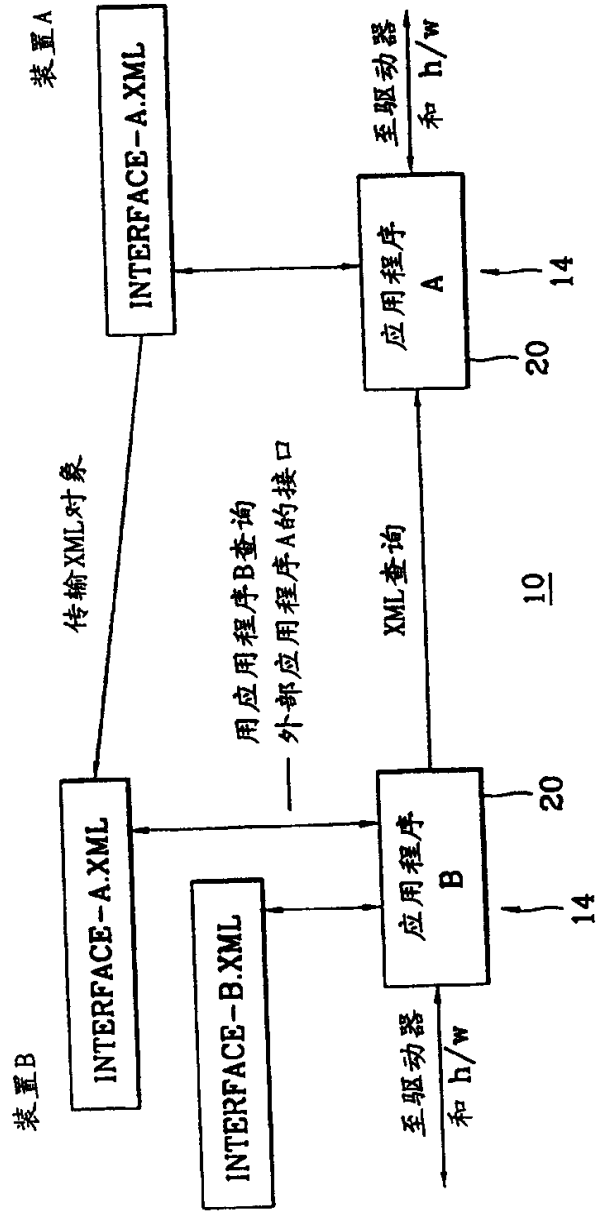
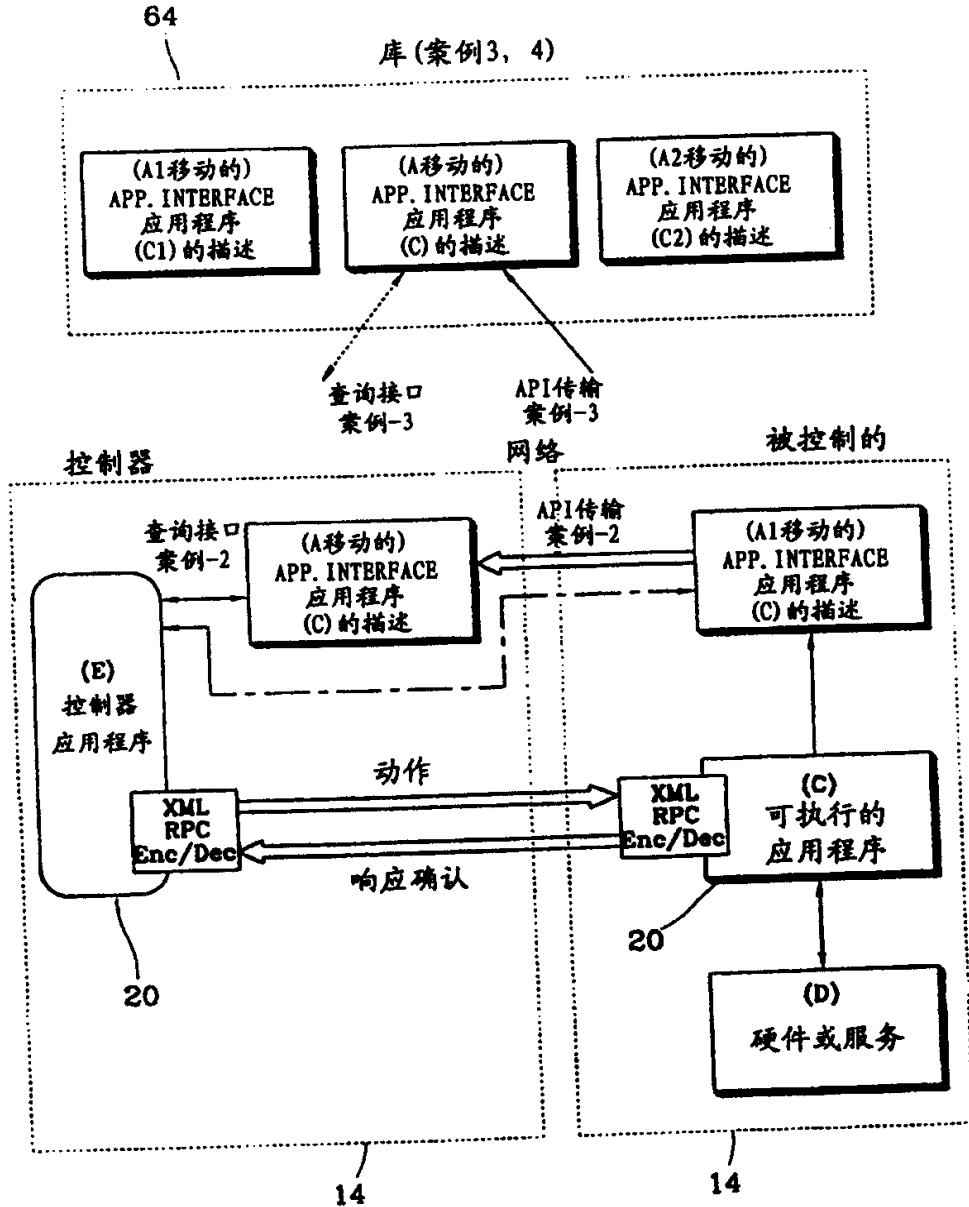


图 16



- 案例1, 用于由 (E) 远程查询的XML应用程序-C INTERFACE (A)
- 案例2, 为了 (E) 的本地查询而将XML应用-C INTERFACE (A) 移动到控制器
- 案例3. 移动XML应用-C INTERFACE (A) 到第3方装置, 该装置为由 (五) 远程查询的全部接口收集地点 (库). 该库必设有可用于直接控制动作和响应的有关应用程序的地址 (URL)
- 案例4, 与案例3类似, 但“直接”控制动作 (和应答) 也直接针对库中的装置. 此时库中建有“传送器”

图 17

XCE/XML协议栈

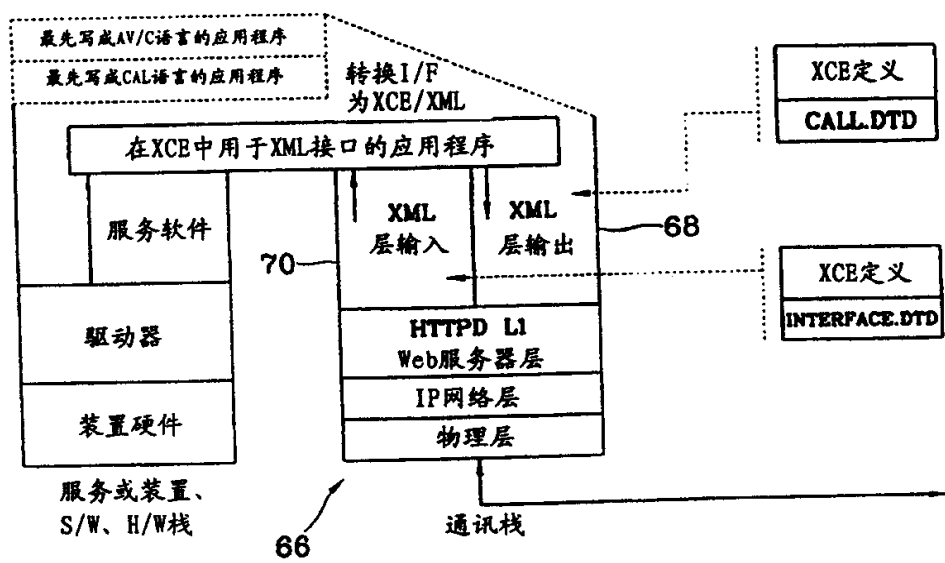


图 18

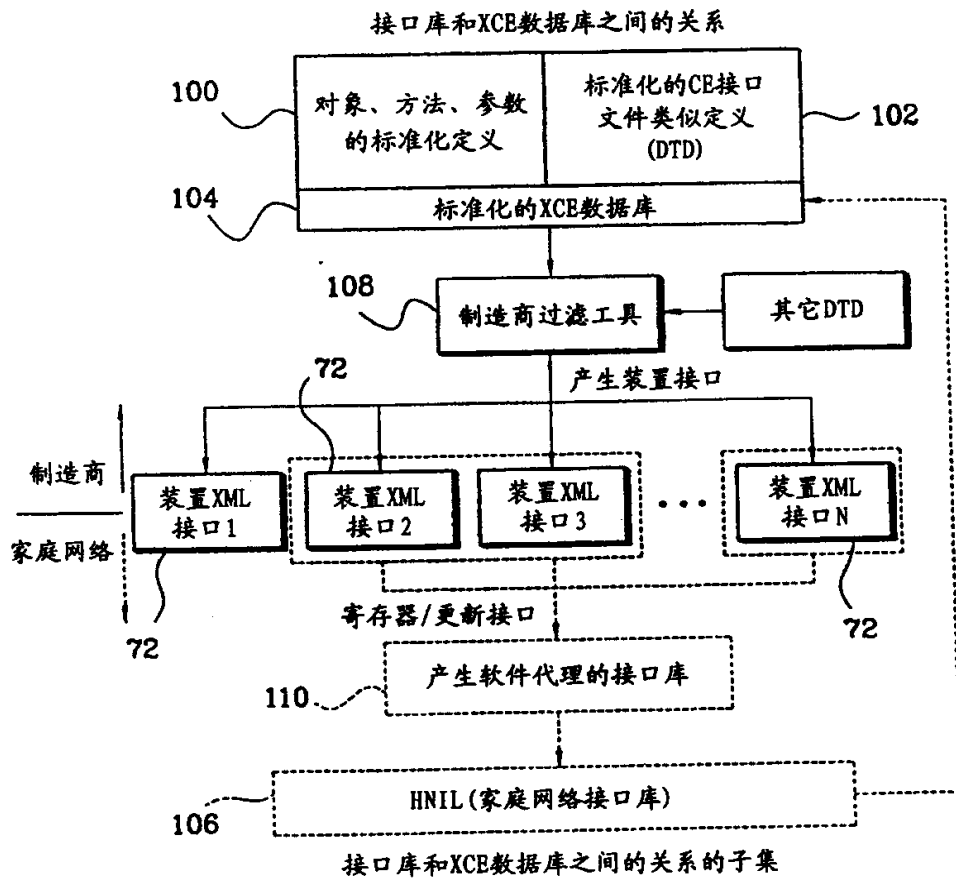
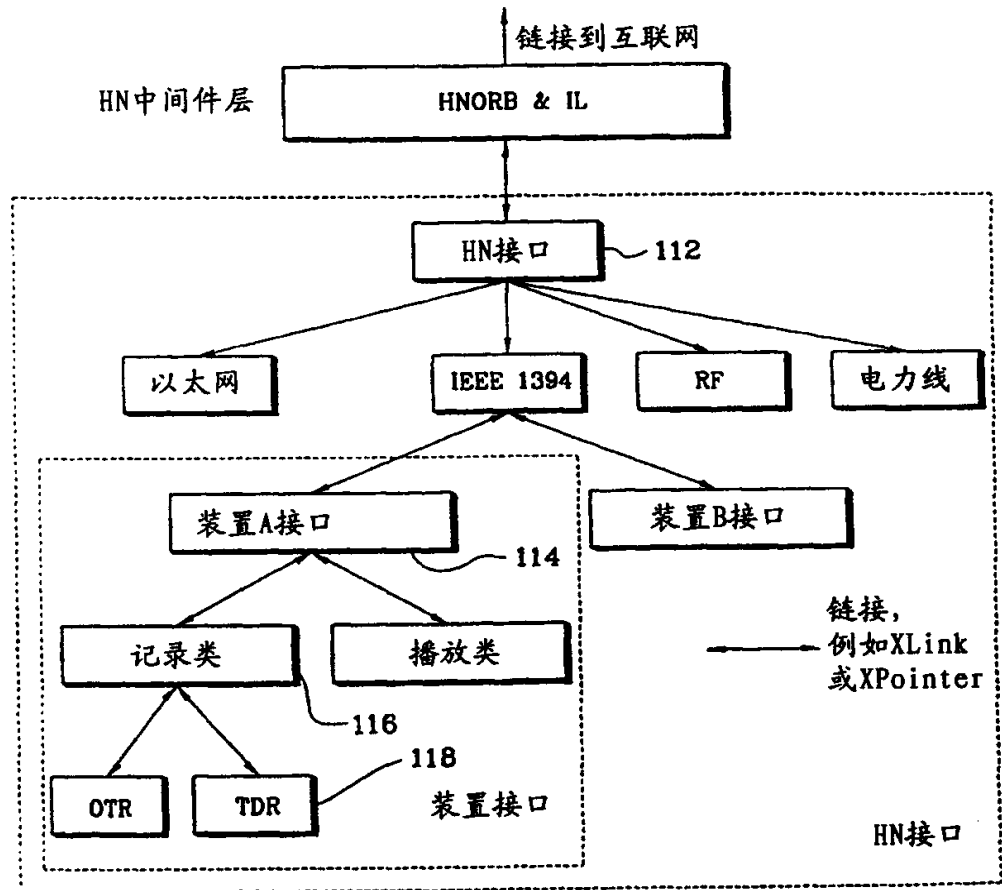


图 20



分层IL和装置接口的示例

图 21

分层接口结构. 四层

层	接口层	接口示例
112 1	用于每个HN的XML接口, 列出当前可用的装置。	MyHN(VCRs, TVs...)
114 2	用于每个装置的普通XML接口, 列出函数类。	普通VCR接口。 (记录、播放...)
116 3	用于一个装置的每个函数类的特殊XML接口。	VCR.record 类
118 4	用于函数类的每个函数的特殊XML接口。	VCR.record.tdr 函数

图 22

0 0 0 0 0

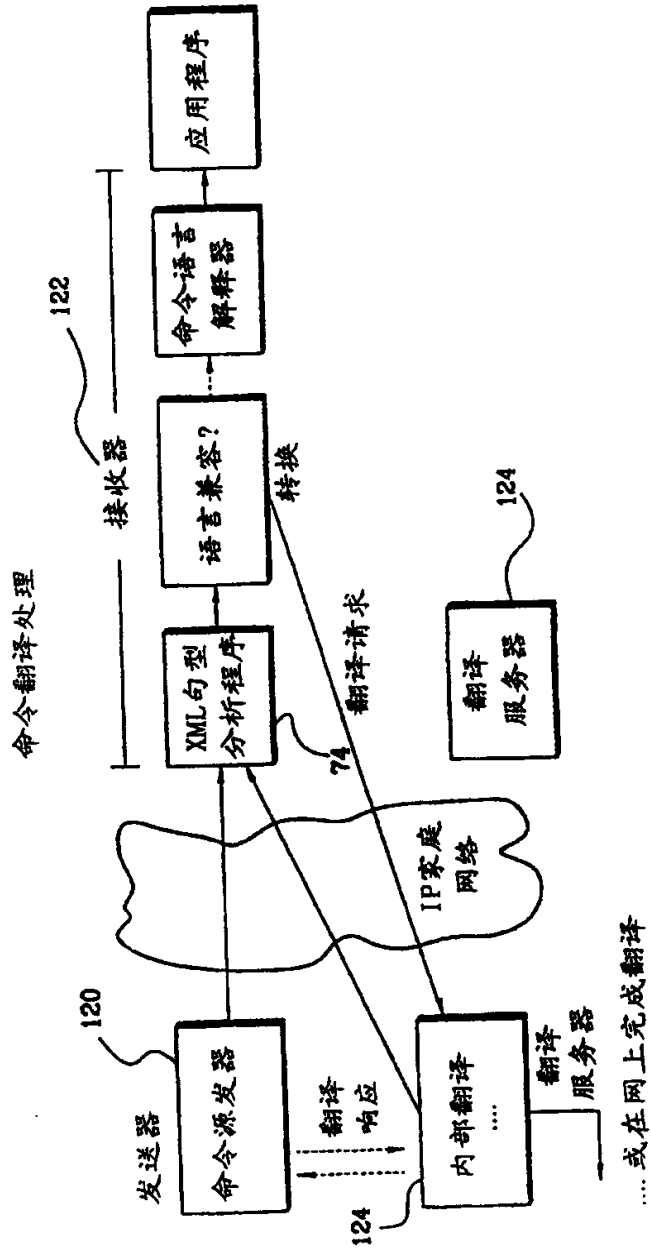


图 23

表3: 示例性格式和类型

类型	格式
命令	CAL, AV/C, X-10
图像	jpeg, gif, bitmap, tiff
语言	英语、法语等
video_stream	mpeg2
video_clib	avi, quicktime, mpeg
文本	html, plain
音频	wav, aiff
应用程序	mword, pdf, postscript, gzip

图 24