

# [12] 发明专利申请公开说明书

[21] 申请号 99811617.3

[43]公开日 2001年11月7日

[11]公开号 CN 1321295A

[22]申请日 1999.10.1 [21]申请号 99811617.3

[30]优先权

[32]1998.10.2 [33]US [31]60/102,957

[32]1999.1.27 [33]US [31]60/117,595

[86]国际申请 PCT/US99/22915 1999.10.1

[87]国际公布 WO00/21073 英 2000.4.13

[85]进入国家阶段日期 2001.3.30

[71]申请人 国际商业机器公司

地址 美国纽约

[72]发明人 斯蒂芬·H·梅斯

[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

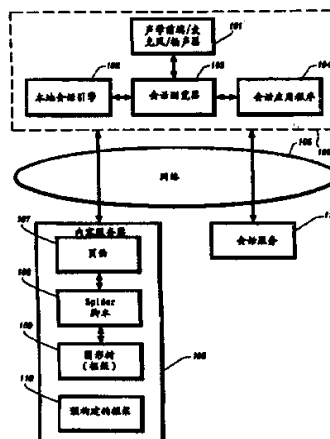
代理人 王以平

权利要求书3页 说明书15页 附图页数2页

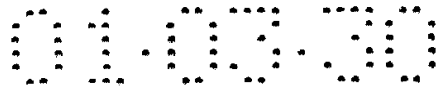
[54]发明名称 通过一般分层对象进行有效语音导航的系统

[57]摘要

一种系统和方法,通过完整描述对话(会话)系统所支持的功能和服务的分层结构(结构框架)来提供快速、有效的会话导航。本发明的提供系统包括:服务器,包含用以产生至少一个分层结构化文档的装置,其中分层结构化文档包括服务器内容的映射和由服务器提供的对话服务的逻辑的映射;以及客户机,包含用以访问至少一个分层文档的装置以及用以激活至少一个分层文档以允许客户机的用户查找由至少一个结构化文档的映射描述的所需内容和对话服务的装置。



ISSN 1008-4274



## 权 利 要 求 书

---

1. 一种用以提供会话导航的系统，包括：

服务器，包含用以产生至少一个分层结构化文档的装置，其中所述分层结构化文档包括所述服务器的内容的映射和由所述服务器提供的对话服务的逻辑的映射；以及

客户机，包含用以访问所述至少一个分层文档的装置以及用以激活所述至少一个分层文档以允许客户机的用户查找由所述至少一个结构化文档的映射描述的所需内容和对话服务的装置。

2. 根据权利要求 1 中所述的系统，其特征在于分层结构化文档是一个包含树结构的框架，树中有多个节点，每个节点都有目标地址，该目标地址包括一个 URL 以及指向由对应节点表示的信息的套接字地址。

3. 根据权利要求 1 中所述的系统，其特征在于每个节点都包含到与节点相关联的 CML - 会话标记语言 - 页面的 URL 的目标地址。

4. 根据权利要求 3 中所述的系统，其特征在于至少一个分层结构化文档由 CML 页面中的一帧表示。

5. 根据权利要求 1 中所述的系统，其特征在于对话服务逻辑的映射包含基本应用程序逻辑的映射以及每个对话服务的对话的逻辑的映射。

6. 根据权利要求 1 中所述的系统，其特征在于至少一个分层结构化文档允许交换过程对象和对话对象，以便向客户机提供对话服务并在客户机上执行对话服务。

7. 根据权利要求 1 中所述的系统，其特征在于每个节点都包含指向负责提供与该节点相关联的对话服务的对话管理器的目标地址。

8. 根据权利要求 7 中所述的系统，其特征在于对话管理器提供基于自然语言理解的表单填写。

9. 根据权利要求 7 中所述的系统，其特征在于对话管理器提供专用的过程。

10. 根据权利要求 7 中所述的系统, 其特征在于对话管理器提供有限状态语法。

11. 根据权利要求 7 中所述的系统, 其特征在于对话管理器利用混合主动性。

12. 根据权利要求 7 中所述的系统, 其特征在于对话管理器利用计算机指导的对话。

13. 根据权利要求 7 中所述的系统, 其特征在于对话管理器利用用户指导的对话。

14. 根据权利要求 1 中所述的系统, 其特征在于, 用于生成至少一个分层结构化文档的装置是“爬行”应用程序。

15. 根据权利要求 1 中所述的系统, 其特征在于分层结构化文档是预构建的。

16. 根据权利要求 1 中所述的系统, 其特征在于分层结构化文档对于客户机的用户是隐藏的。

17. 根据权利要求 1 中所述的系统, 其特征在于只要分层结构化文档处于活动状态, 用户随时可对其进行询问。

18. 根据权利要求 1 中所述的系统, 其特征在于分层结构化文档可在对话开始时被用户询问。

19. 根据权利要求 1 中所述的系统, 其特征在于客户机是变码器, 分层结构化文档用于通过变码器执行逻辑变码, 从而生成 HTML 页面和 CML 页面。

20. 根据权利要求 19 中所述的系统, 其特征在于分层结构化文档代表 HTML 站点的结构。

21. 根据权利要求 19 中所述的系统, 其特征在于分层结构化文档可以是多形态、仅限于语音和仅限于 GUI 之一。

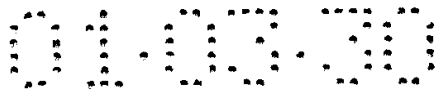
22. 根据权利要求 1 中所述的系统, 其特征在于分层结构化文档是一个对话组件。

23. 根据权利要求 1 中所述的系统, 其特征在于每个节点都包含到至少一个对象的目标地址。

24. 根据权利要求 1 中所述的系统，其特征在于分层结构化文档可通过程序和陈述这两种方式之一实现。

25. 根据权利要求 1 中所述的系统，其特征在于客户机和服务器都在同一计算机上执行。

26. 根据权利要求 1 中所述的系统，其特征在于分层结构化文档存储在服务器、客户机、网关和路由器之一上。



## 说 明 书

---

### 通过一般分层对象进行有效语音导航的系统

本申请基于下列两个临时申请：美国序列号 60/102,957 (1998 年 10 月 2 日申请)以及美国序列号 60/117,595 (1999 年 1 月 27 日申请)

总体而言，本发明涉及会话系统，更具体地讲，是提供快速、有效的会话导航的系统和方法；会话导航的对象可以是特定站点（例如，Web 站点）的 CML（会话标记语言）页面或使用框架或树结构的会话服务（在用户首次连接到服务提供商时，框架或树结构描述该站点、站点的一部分或服务），或者与给定的服务相关联的复杂对话。

在不久的将来，诸如 NLU（自然语言理解）和 NLG（自然语言生成）以及对话管理之类的会话系统将成为人/机界面的重要组成部分。为使这些会话系统功能更强，这些系统应允许用户尽可能自然地提出查询，就象在普通对话中那样。的确，此类系统应提供混合或定向的主动性，以使用户能够基于系统的反应完成、纠正、修改或澄清自己的查询，同时还应提供利用查询快速访问复杂分层菜单的能力。

然而，自然对话系统常规实例化的设计方式使得用户在组成其查询的序列或元素中要使用限制性相对较强的逻辑。对于 FSG（有限状态语法）或纯统计 NLU 系统就是如此。确实，NLU 十分注重理解对于给定的环境、应用程序或查询模式的输入信息。

而且，对于会话系统，当语音输出为仅有的输出形式时，便很难向用户提供复杂的服务（例如，Web 页面、分层的 DTMF 菜单、有多个选项的 NLU/NLG 系统以及基于表单的实现）。通常，用户必须浏览多个 Web 页面或 DTMF 菜单才能找到感兴趣的信息。因此，极需一种系统和方法来提供必要的信息，从而使用户能够快速浏览整个站点或可用的对话服务，而无须查看或收听指向所需服务或交易的每个可能的页面或菜单。

本发明涉及一种系统和方法，用以通过完整描述对话（会话）系

统所支持的功能和服务的分层结构（结构框架）来提供快速、有效的会话导航。

一方面，提供会话系统和方法，将对话菜单和目标地址预加载到与其相关联的对话管理过程，以便处理多个或复杂的模式、环境或应用程序。例如，内容服务器（Web 站点）可以下载描述用户初次与服务器相连时由服务器提供的内容（页面）或服务的框架或树结构。该框架对用户是隐藏的（不讲话），但用户可以发出语音命令来访问感兴趣的页面或特定的对话服务，其中，语音命令通过反应正常（按照用户命令）的会话系统使用框架中所含的信息进行识别。

本发明特别适用于仅以语音作为输入形式的系统，原因是它允许频繁使用系统的用户采用快捷方式来会话导航多个页面、对话组件或 NLU 对话表单，以便找到感兴趣的信息。该框架提供了必要的信息，使得用户能够快速浏览整个站点或可用的服务，而无须查看或收听指向所需服务或交易的每个可能的页面。

本发明的上述信息以及其它的方面、功能和优点将在下列关于优选实施例的详细说明中介绍并逐渐变得明朗，阅读说明时应参照附图。

图 1 是一个方框图，是使用依据本发明实施例的会话框架提供会话导航的系统；以及

图 2 是一个流程图，是使用依据本发明一个方面的会话框架提供会话导航的方法。

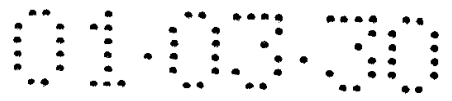
应理解，本发明可以通过各种形式的硬件、软件、固件、专用处理器或这些内容的组合来实现。优选情况下，将本发明实现为一种包含确实体现在程序存储设备上的程序指令的应用程序，且能够由包含适当结构的任何设备（嵌入的或以其它方式）或机器执行；上述的程序存储设备可以是软盘、RAM、CD-ROM、ROM 和快闪存储器等。还应理解，由于附图中显示的某些系统组成部件和方法步骤优选情况下是以软件实现，系统部件（或处理步骤）间的实际连接会根据本发明的编程方式而有所不同。有了此处的说明，相关技术中的一个普通技术人员便能够设想对于本发明的上述或者类似实现或配置。

还应理解，本发明可以在任何基于会话/对话的系统中实现，其中，用户为了获得所感兴趣的信息或服务，可能不得不收听很多页面、菜单或对话组件或者按层次结构浏览一系列页面、菜单或对话组件。尽管本发明可以由此类系统使用，但为便于说明，本发明将对照会话浏览器和 CVM（会话虚拟机）系统（公开于存档号为 IBM YO 998-392P 的专利申请“会话浏览器和会话系统”以及存档号为 IBM YO 999-111P 的专利申请“通过会话虚拟机进行会话计算”中，这两个专利是同时申请、共同转让的并完全列于此，以供参考）进行阐述。

上述的申请 IBM 存档号 YO 998-392P 讲述的内容是，利用会话浏览器和 CML（会话标记语言），使用以 CML 编写的一系列相互链接的页面来构建会话系统。这个全新的概念类似于使用以 HTML（超文本标记语言）编写的一系列相互链接的页面来构建常规的 GUI 浏览器应用程序。此外，正如 HTML 可提供一系列机制，将 GUI 操作转换为应用程序操作（如，访问其它页面或与服务器进行通信），会话浏览器和 CML 用于将语音输入转换为类似的应用程序操作。

特别指出，CML 页面描述了要通过会话浏览器解释且呈现给用户的会话 UI。在一个实施例中，CML 页面是通过基于当前 XML（可扩展标记语言）规范的标记得以限定的。主要元素包括 <page>、<body>、<menu> 和 <form>。Page 元素集合了其它 CML 元素，是 CML 文档的顶级元素。Body 元素指定要由浏览器输出的语音。Menu 元素会给用户若干个选项，并将每个选项与一个目标地址（例如，URL 和套接字地址）关联起来，该地址标识了为用户选择该选项时要访问的 CML 元素。Form 元素允许用户提供一条或多条信息，其中各信息的内容例如均由语法进行描述。此外，对话对象（如，小应用程序）可以加载到页面中。而且，NL 对话的多个同时表单可以根据需要加载到 CML 页面中。这种情况下，描述内容可以只使用语言模型以及标记和语法分析参数而不使用语法，也可以二者都使用。Form 元素还指定当用户完成表单时要访问的目标地址。

从上述参考信息中更为完整的讲述可知，在使用会话浏览器的纯



会话系统中，语音和音频是提供给用户的唯一输出。如果用户希望如此，将从服务提供商下载 CML 页面。然后，会话浏览器使用 CML 页面中的文本到语音标记语言信息向用户读出该页面并收听用户发出的命令，以便查询链接、填写表单或操作浏览器；这里的语音命令由语音识别程序（使用 CML 页面语音识别部分的线索）进行解码。

这个会话界面意味着，为了能访问到感兴趣的信息，用户通常可能需要收听大量 CML 页面或者按层次结构浏览一系列 CML 页面。如以上 IBM 存档号 Y0 998-392P 中所述，使用会话浏览器进行导航的一种方法是“所闻即所讲”方法，用户可通过引用已经向用户读过的页面和链接来使浏览更快捷。此外，还有另一种导航形式是激活已加载页面上的所有链接，即使这些链接尚未读给用户也不例外。然而，每种方法都要求用户浏览连续的页面，以便访问在第一页上没有链接到的页面。下面将对照图示的实施例介绍另一种方法。

参照图 1 的方框图，它展示了依据本发明的实施例提供有效的会话导航的系统。系统包含客户机 100，它可以是个人计算机、智能电话、PDA（个人数字助理）或任何支持语音功能的普及计算（PvC）设备。客户机 100 包含声学前端/麦克风/扬声器 101，用以处理音频/语音输入以及输出由会话浏览器 103 生成的音频/语音。客户机 100 还包含一个或多个本地会话引擎 102，用以处理由声学前端 101 生成和/或捕获的声学特性和/或波形，以及根据会话浏览器 103 的请求和管理而生成语音输出。本地会话引擎 102 例如可以包括嵌套的语音识别引擎、扬声器识别引擎、TTS（文本到语音）引擎、NLU（自然语言理解）引擎和/或 NLG（自然语言生成）引擎。会话浏览器 103 与一个或多个支持语音功能的本地应用程序 104 交互，以便向用户提供有关此类应用程序的语音输出。

如图 1 的示例所示，客户机 100 通过网络 105 与包含一个或多个“站点”（例如，CML 页面 107）的远程服务器 106 联网。应理解，术语“站点”此处指的是特定 CML 页面上相关 CML 文件（或页面）的集合。此外，尽管图 1 中只显示了一个服务器 106，但应知道，特



定的“站点”可能延伸到一系列在不同位置上连接到网络 105 的其它服务器（未显示）。因此，为便于说明，假定与特定的“站点”相关的所有页面均位于相同的服务器 106 上。

此外，应理解，会话引擎 102 可以分布在网络 105（尽管在图 1 中它们只作为客户机 100 的本地引擎）中并可使用在同时申请的并公开于存档号为 IBM YO 999-113P 的专利申请“提供网络协调会话服务的系统和方法”（这个专利是共同转让的，列于此处，以供参考）中介绍的技术进行访问。此概念如图 1 中的会话服务服务器 111 所示，该服务器可以提供任何形式的会话服务。尽管图中只显示了一个服务器 111，但会话服务服务器 111 可以代表多个服务器。而且，会话服务器 103 可以位于网络上并由客户机 100 访问。

依据本发明，内容服务器 106 包括 spider（蜘蛛）脚本 108（如行业中所知），用以在客户机 100（通过会话浏览器 103）访问服务器 106 之前或之时动态生成图形树或框架 109。这里，框架 109 可以是有分层结构的文档（文件），描述了在用户初次与服务器 106 相连时下载的所请求的“站点”或该“站点”的一部分。特别是当客户机 100 初次与内容服务器 106 建立网络连接时，服务器 106 将执行 spider 脚本 108（即，在行业中称为“crawler”、“robot”或“bot”的应用程序），spider 脚本 108 会“爬过”所请求 CML 页面中的每个链接以提取信息（例如，从由作者提供或者通过自动主题提取的标题、元信息）并将其存储在图形树结构 109 中，从而用相对的已提取元信息映射该页面。另外，服务器 106 可以提供由作者为服务器 106 中每个特定页面 107 预构建的框架 110。由此，可以提前执行上述的“爬行”，从而使框架能够在与服务器连接后立即传送给服务器，而不必由于等待执行“爬行”而发生延迟。

应理解，可以使用除常规“爬行”以外的方法来实现“爬行”功能。例如，替代常规方法的一种方法是通过协议调用，令会话浏览器（或基本平台，例如，CVM）与 Web 服务器相连并从由服务器提供的映射中获得信息。另外，它也可以从存储在网络中其它位置的其它高

速缓存/存储功能获得这些信息，以便在网络上提供适当的呼叫路由和通信量/负载平衡（如 Y0999-113P 所述）。

应理解，“爬行”可以由内容服务器 106、会话服务服务器 111 或者会话浏览器 103 执行。优选情况下，提前执行“爬行”。的确，尽管客户机 100 或者服务器 106、111 都可以在与站点连接时生成“爬行”，但是由于对框架的访问（进而对话初始化）会延迟，这并不象提前执行“爬行”那样实用。因此，服务器 106 可以定期地在站点上“爬行”并存储得到的框架，以便能够在建立连接时将其发送给客户机 100。对于电话浏览器（其中会话浏览器位于远程服务器上），由于服务器会话浏览器（或者浏览器平台，例如，CVM）可以访问的站点通常很有限，浏览器可以替代服务器 106 执行“爬行”。

还应理解，尽管图 1 的示范实施例显示服务器 106 与客户机 100 联网，但取决于应用程序，服务器功能也可以包含在应用程序或对话中，这里的应用程序或对话从对于客户机 100 为本地的资源执行或来自该资源，且客户机 100 与服务器 106 可能位于同一计算机上。此外，该站点可以是实际站点、对话或应用程序（本地或远程）。还应进一步认为，框架 109 的存储（和访问）可在网络 105 中任何计算机上进行，如，客户机 100、服务器 106、网关或路由器。

在说明性实施例中，框架（109 或 110）以 CML 文件形式提供（尽管应理解框架可以作为与陈述形式（例如，XML、CML）相对的过程输入而提供）。应理解，无论怎样实施（是 CML、任何其它陈述性机制、过程等），由结构框架为传递信息而执行的一般功能都保持不变。这些信息包含与应用程序或站点相关联的对话的不同组件的分层菜单结构；每个对话菜单级别上的选项；和/或构成每个菜单的对话组件（以下将逐一进行介绍）。

框架（109、110）的每个元素均使用隐藏标记标出，以使会话浏览器 103 不向用户讲出这些元素。例如，CML 文档的框架部分可以使用 `<skeleton></ skeleton >` 标记或任何等同的方式标记出来。框架的页面通过站点目录结构以常规的路径形式（前缀、中缀等）提供。

以下是 CML 文件格式中框架的一个示例：

```

<skeleton>
<Hidden>
<Menu>
<a target="target address 1">politics</a><Menu>
  <a target="target address 2">International</a><Menu>
    <a target="target address 3">Europe</a><Menu>
      <a target="target address 4">Belgium</a><Menu>
        <a target="target address 5">France</a>
          </Menu>
        <a target="target address 6">Africa</a>
          </Menu>
      <a target="target address 7">US</a>
    </Menu>
  <a target="target address x">business</a>
  <a target="target address y">weather</a>
  <a target="target address z">sport</a><Menu>
    <a target="target address z2">International</a><Menu>
      <a target="target address z3">Europe</a><Menu>
        <a target="target address z4">Belgium</a>
        <a target="target address z5">France</a>
          </Menu>
      <a target="target address z6">Africa</a>
        </Menu>
    <a target="target address z7">US</a>
  </Menu>
</Menu>
</Hidden>
</skeleton>

```

显而易见，以上的框架结构代表了一个在每个节点上都有 target（目标）的分层树结构，用以完成相关联的查询。例如，锚标记 <a>...</a> 与属性“target”一同使用可以创建一个“链接”，如在本行业中所理解的。每个“target address”（目标地址）可以是 URL 或套接字地址或者任何可从中获得对话或对话的一部分/参数的地址。例如，目标地址可以是 HTTP 协议中的 URL，也可以是会话浏览器应用程序。基于上述示例，访问 CNN 站点的用户可以立即访问有关“Belgium”（比利时）的“international”（国际）新闻，或者请求有关货币兑换汇率的任何服务，而不必辛苦地浏览以下页面：news（新闻）、international（国际）、Europe（欧洲）、Belgium（比利时）等。

应认识到，在上述模型中，任何目标均可通过引用地址来访问如下内容：

1. 会话浏览器的 CML 页面（例如，URL）
2. 驱动特定模式/服务的对话的过程（例如，小应用程序或 CGI 脚本、ActiveX 组件）或者完成用户查询的一般过程（如图 1 所示的网络 105 上的会话服务 111）。例如，服务可以包括用于音频捕获、CODEC 压缩和通信协议的本地插件、用于分布式处理的基础设施以及服务器端对应的元素（如 Y0999-113P 中所述）。另外，可以加载一个过程来执行对话，而不必依靠 CML 页面实现对话。这可以是直接与不同的会话引擎交谈以构建对话的任何程序，也可以是对话组件、基础类别对象的组合（如以上 Y0999-111P 中所述）以及由基本平台（例如，CVM）提供的服务。的确，过程可以由 NLU/NLG、FSG 驱动，主动权由用户（用户需要构建查询）、系统（系统不断地提出问题，以补充使查询符合条件的所有参数）掌握，也可使用混合主动权：用户和系统交互来完成非模糊查询。更具体地说，混合主动权就是由用户和系统共同驱动对话。用户可以在不被计算机提示的情况下发出有关哪些内容被激活的查询。此外，用户提供的答案可以比计算机提出的问题多或者答非所问。如果对话支持用户输入的域，而且已经激活了他的查

询所请求的功能（例如，有对应的表单），则对话将切换以完成、纠正和/或澄清该查询。对于由计算机指导的对话，计算机驱动对话，即，用户必须回应特定的提示。根据本发明，用户可以激活框架以跳转到另一个提示并从该处继续进行由计算机驱动的对话。对于由用户指导的对话，用户驱动对话的方式是：进行查询，直到查询完成且已无歧异为止，然后计算机执行这些查询（类似于在 HTML 中填写表单，即，用户填写表单，填写满意后提交该表单）。

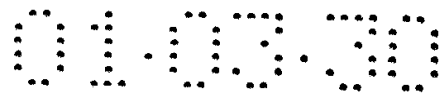
3. 向用户提供的一系列表单，对于给定节点指示的服务，通过基于表单的 NLU/NLG 来完成；以及

4. 语音（波形、特性）被送去处理的位置（如以上申请 IBM 存档号 Y0999-113P 中所述）。

表单在以下参考资料中有说明：Kishore, et al. "Free-Flow Dialog Management Using Forms," Proc. Eurospeech 1999, Budapest Hungary, September 1999 和 Davies and al., "The IBM Conversational Telephony System For Financial Applications," Proc. Eurospeech 99, Budapest Hungary, September 1999, 这些资料列于此处，以供参考。

因此，如上所述，框架可以发出框架专用的对话（无论隐藏与否）。这意味着，可以加载表单或对话组件（会话对象），以便允许用户利用框架功能（而不是通过 CML 隐藏组件）。框架可以引用新表单、CML 页面或过程对话。框架也可以作为 CML、多余表单或过程对话组件而提供。

而且，如上所述，本发明支持从站点自动提取框架，以便映射交易的逻辑或特定站点的内容。此外，框架可以由站点开发人员/业务逻辑所有者或内容提供者来开发或编写。应理解，编写/开发框架时，应捕获应用程序的逻辑以及对话的逻辑（应理解，对话的捕获可以有多种形态，即它可以包含 GUI 和语音成分（同步）或其它形态，或者仅限于语音或 GUI）。对话捕获不仅限于 CML 方法。的确，当会话浏览器或任何其它会话应用程序写在 CVM 之上时（如以上 IBM 存档号



Y0998-392P 以及 Y0999-111P 中所述)，对话的隐藏部分仍应被捕获，无论它是应用程序的完整业务逻辑还是对话组件之后的较短的逻辑。

因此，在过程的情况下，应用程序可以通过与类似于此处所述的示范 XML 框架结构内容（通常带有一个定义数量和字段性质的标题）的交换对象结构的方式来交换业务逻辑信息。对象的这种过程性（例如，RPC（远程程序调用））交换可以承载更多信息。的确，这些对象还可以包含整个对话管理器、对话逻辑以及对话模块的环境（即，对话的一部分）。有了所有的引擎调用和 UI 组件，这可以是对话的完全（full-blown）实现。例如，它可以是旨在收集电话号码的完全基于语法或完全基于 NL 的对话。另一选择是提供一系列实现基本对话组件的会话基础类别。这些会话基础类别和对象在以上的 IBM 存档号 Y0999-111P 中进行阐述。

复杂对话的构建可以通过串行或并行组合这些基础类别而实现。串行组合使用与实现额外 UI、行为和对话以及执行逻辑操作的与代码交错的组件。并行组合则将不同的对话管理作为同时发生的应用程序来激活。由 CVM 提供的基本对话管理器决定活动的对话。CVM 提供基础类别的库。因此，在下载时，这些类别不进行传送，而是链接在浏览器或 CVM 平台上。

而且，对于会话浏览器的情形，有同步范围的表单（如以上 IBM 存档号 Y0998-392 中所述）以及小应用程序、插件和其它对话模块可被视为展示完整对话逻辑的框架示例。

因此，框架可以是基于 XML（或其它陈述技术）或者过程（或各过程的组合）的任何结构，能够提前捕获和激活部分对话，否则在用户在对话中导航来激活该部分对话之前，该部分对话一直被对话逻辑或业务逻辑隐藏。框架驱动的对话可以通过陈述平台（例如，会话浏览器）或过程平台（例如，对话组件等）作为应用程序对话的一部分被激活。此应用程序对话本身可以基于过程形式或陈述形式或它们的任意组合。

因此，应理解，框架的实现可以与 CML 实施例不同：功能（即，

要通过框架提供的信息)可以通过其它方式(如,被交换的对象)实施。此外,对这类信息的使用也可以通过其它方式实现,例如,通过对话的过程定义或对话的陈述定义的其它形式。

下例(IBM 存档号 Y0998-392P 中有详述)展示使用表单与过程的组合来激活复杂对话的整个逻辑:

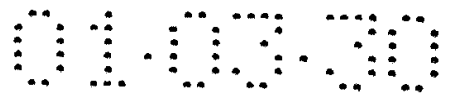
```
<menu name=main1 scope=global>
<form name=skeleton_form scope=multiple context=maintain></form>
<form name=form1 load=procedure1 scope=mulitple context=maintain></form>
<form name=form2 load=form2 scope=mulitple context=reset></form>
<form name=form3 load=procedure3 scope=deactivated context=reset></form>
<form name=form4 load=procedure4 scope=mulitple context=reset></form>
...
</menu>
```

上述示例表明,依据本方法可将框架表单加载在对话中。

因此,依据本发明,有结构的框架可以通过交换框架 XML 信息、框架对象或者通过预加载和激活对话组件,来捕获对话的逻辑或者站点或应用程序的业务逻辑。这些对话组件可通过预加载 XML 页面、预加载会话(过程)对象或这些内容的组合来提供。有同步范围表单的机制便是如何与会话服务器一同实现此目的的一个示例。

通过预加载和激活对话组件,可加速对给定服务或者交易的某部分的访问,而无须遵循服务提供商最初提供的顺序、回放或菜单。这几乎等同于有操作员的对话,其中用户对交换这样的信息不感兴趣:操作员首先尝试获得但是立即转接到给定的服务、立即提供执行特定的查询或立即返回先前的交易以便继续进行先前交易所需的最小信息。通过激活整个对话框,系统能够填写与该查询相关联的表单,而且在提供了足够的信息时能够继续并完成查询。如果查询未完成,对话系统将只尝试收集完成该表单所需的额外信息。

参看图 2,这是一张流程图,展示了依据本发明的一个方面提供有效的会话导航的方法。应理解,为了便于说明,此处介绍的方法基



于使用 CML 和会话浏览器的系统。起初，客户机将（通过会话浏览器）连接到服务提供商的服务器，以访问特定的 CML 页面（站点）（步骤 200）。如果所请求的站点有预构建的框架（步骤 201 的结果是肯定的），预构建的框架将被下载到客户机（步骤 203）。

另一方面，如果没有预构建的框架（步骤 201 的结果是否定的），会自动生成站点框架（步骤 202），然后将其下载到客户机（步骤 203）。应理解，该框架可代表整个站点或站点的一部分。此外，框架还可以代表整个对话（例如，实现整个对话的过程）或对话的一部分（例如，描述对信用卡、电话号码或电子邮件客户程序进行访问所需的整个组件，处理日历项、订票、查看日程等所需的整个对话组件等）或者它们的组合。因此，理想情况下，用户可以在信用卡或地址填写对话过程中直接跳转（这可能是在购物完成之前）。此外，在执行另一个对话（如，银行交易或电子邮件撰写）时，用户可以跳转到日历查看对话中。因此，通过捕获和激活其它对话部分（而不仅是站点信息），系统可以让用户在执行给定的任务/对话的同时访问其它应用程序或应用程序的部分。应理解，这可以使用 CML 实现来执行，其中每个对话部分均可通过特定的 CML 页面或由这些页面加载的过程来实现。还应理解，站点框架可以由作者预构建（步骤 201），也可以通过令站点预“爬行”的方式预构建（如上所述）。

一旦框架被下载（步骤 203），会话浏览器将激活框架中包含的所有页面（链接）（204）。起初，尽管页面链接被激活，但框架对用户仍然是隐藏的（即，不讲话）。然而，如果用户发出语音命令激活特定的链接以访问特定的页面（用户认为它与起初下载的页面相关）（步骤 205 的结果是肯定的），会话浏览器将分析框架的语法以找到框架中与识别出的用户命令相对应的链接（活动文档）（步骤 206）。如果找到与用户命令对应的已激活链接（步骤 207 的结果是肯定的），将下载该站点（步骤 208）并重复该步骤（返回步骤 208）。此功能特别适用于以语音作为唯一的输入/输出形态的系统（它也适用于多形态系统）。熟悉特定站点（及其相关站点）的用户可以快速导航一系



列已知的 CML 页面，而无须查看和收听指向所需页面、服务或交易的每个可能页面。

另外，用户可以询问会话浏览器，以指明框架中指向所需主题或服务的特定链接是否与用户查询相匹配，或者通过让浏览器读回框架中的信息来浏览框架（步骤 209 的结果是肯定的）。当用户要查看给定的主题或者要收听基础页面时，浏览器将搜索框架并读回信息或识别查询匹配情况（步骤 210）。一旦浏览器提供所请求的信息，用户便可以（在框架中）选择所需的链接（步骤 211），以访问所需的站点或服务（步骤 208）。本发明的这个方面特别有用，原因是框架不仅可以包含有关对话的信息，还可以直接包含对话，以使用户能够搜索并跟踪该信息（适用于用户不熟悉对话或者应用程序或站点太复杂以致需要混合主动性对话方可找到适当的快捷方式或要加载的组件的情形）。

更为理想的是，“站点”框架可以扩展为包含与可视（GUI）浏览器中的框架等价的音频内容。特别指出，可由用户导航或查询的额外隐藏信息是由页面作者编辑，并由作者手工构建且作为框架添加进去（即，与主 CML 页面一同加载的单个页面）的。在 HTML 中，框架允许用户将页面分成不同的内嵌子页或带有指向不同 HTML 页面的目标地址的子页。同样，对于 CML 页面，也可以引入框架来向用户展示菜单或框架。例如，作者可以按下例显示其站点页面：

```
<CML>
  <frameset interface="hidden,play"><!这意味着第一个框架是隐藏的，第二个则
播放给用户>
  <frame 1>
  <skeleton target=url1>
  </skeleton>
  <frame 2>
  ...[这一阶段对话的主 CML 内容页面。也可用 target 标记指向它]
  </frame 2>
```

</frameset>

</CML>

这是激活对话的另一个示例。前一个示例展示了使用以并行方式激活的多个表单，此示例则展示了一种不同的用于并行激活的语法。

显而易见，本发明适用于各种功能。在上述的专利申请 Y0999-111P 和 Y0998-392P 中介绍了以下概念：“SpokenAge”是应用程序开发环境，“会话变码”是将常规演示材料转换为会话材料（例如，将一般 HTML 页面转换为 CML 页面）的机制。

这两个申请中还介绍了功能变码的概念：将演示信息转换为 CML（即，将给定的 HTML 页面转换为 CML 页面）以及“逻辑变码”的概念。如上述申请 Y0999-111P 和 Y0998-392P 所述，逻辑变码涉及基于站点后台的业务逻辑来构建对话。以前，由于变码器无法访问有关对话的“下一步”以及交易业务逻辑的“下一步”的信息，因而此方法非常困难。

由于结构框架可以捕获业务逻辑（多形态、仅限于语音或仅限于 GUI），因而与逻辑变码一同使用结构框架会十分有利。当然，这时的框架是 HTML 站点（而不是 CML 站点）的结构，但该框架的基本原理并未更改。因此，应理解，结构框架的概念不局限于语音对话，而是可扩展为任何对话，即使是基于 GUI 的对话也不例外。因此，通过访问对话/交易或站点的框架并将其展开显示给应用程序开发人员（作为站点图或对话图）或自动化的变码器，便可提供逻辑变码工具。同样，可以对变码后的 CML 对话进行优化，以包含现在可用的所有业务逻辑信息。

更为理想的是，本发明也适用于与多形态会话浏览器一同使用。的确，在多形态情况中，GUI 内容是作为常规 GUI 浏览器设计的这一点十分明确。的确，真正引人注目的多形态页面更倾向于使用 GUI 页面来向用户显示某些只通过语音很难表达的信息。这还要求预加载框架，以增强对站点的会话导航功能。

在本发明的另一个说明性实施例中，可以考虑涉及帐户管理、联

机支付和投资管理的 NLU/NLG 联机银行系统。例如，可以使用与由用户执行的所有操作相对应的几个表单来构建此系统。可以实现 NLU 语法分析和转换，以便提供表单的所有输入字段的参数；表单与参数和环境兼容。可通过在完成前一个查询之前激活新表单（即，查询）趋势的方式嵌入查询。在完成嵌入查询后，可通过查看尚未完成的先前表单来重新激活前一个环境。

如果能够提供的查询的可能类型有所增加，并行表单填写会变得困难。以往，用户需要（通过 NLU 查询和混合主动性）填写一系列表单方可定义自己的意向。本发明的有利之处在于使用框架来代表更高级别的菜单，它可以表示为要为在给定的层次级别上填写的每套表单添加的一个额外表单。因此，频繁使用系统的用户现在可以（通过查询）激活框架的所需节点来跳过串行表单填写并立即缩小查询范围。

可以想到，本发明可用其它各种应用程序实现，以便提供有效的会话导航。这些系统的示例包括：

- PVA（个人车辆助理），这里会话浏览器是主 UI 环境；
- VA（虚拟助理），这里 NLU/NLG 对话可以由表单、FSG 或过程驱动；
- 有会话浏览器的 PDA 和智能电话；
- 企业软件解决方案和电子商务，以及电话应用程序，这里 IVR（交互式语音响应）脚本被 CML 和会话浏览器架构所代替，上述两项内容都在上述申请（IBM 存档号 Y0998-392P）中有所阐述；以及
- 电话/电话亭（kiosk）应用程序，可在其中应用基于表单的 NLU 工具箱。

尽管此处参照附图介绍了说明性实施例，应理解，本系统和方法不局限于这些实施例本身，本领域内普通技术人员可能想到不背离本发明的范围和宗旨的各种其它更改和修正。所有这些更改和修正都应包括在权利要求书限定的发明范围内。

说明书附图

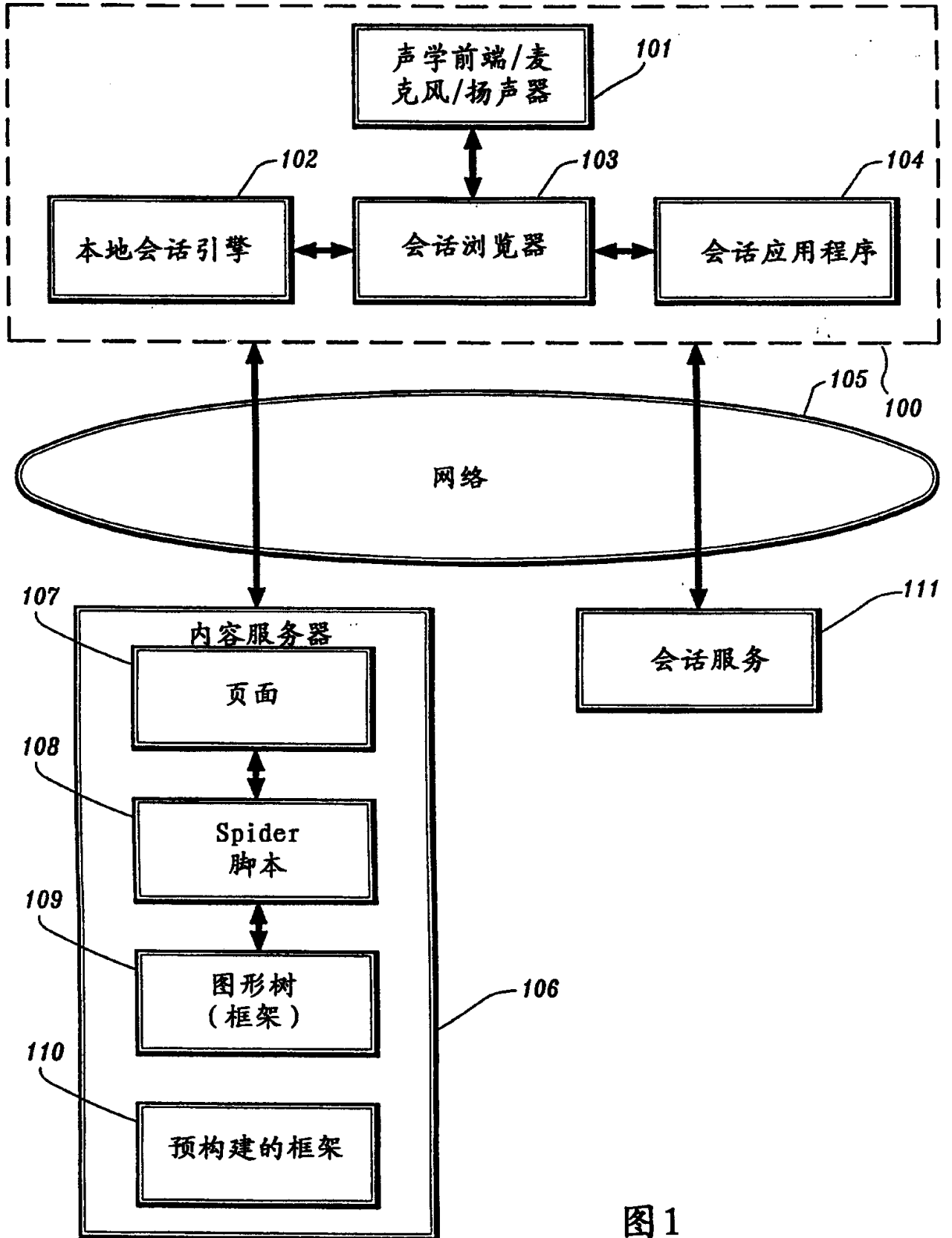


图1

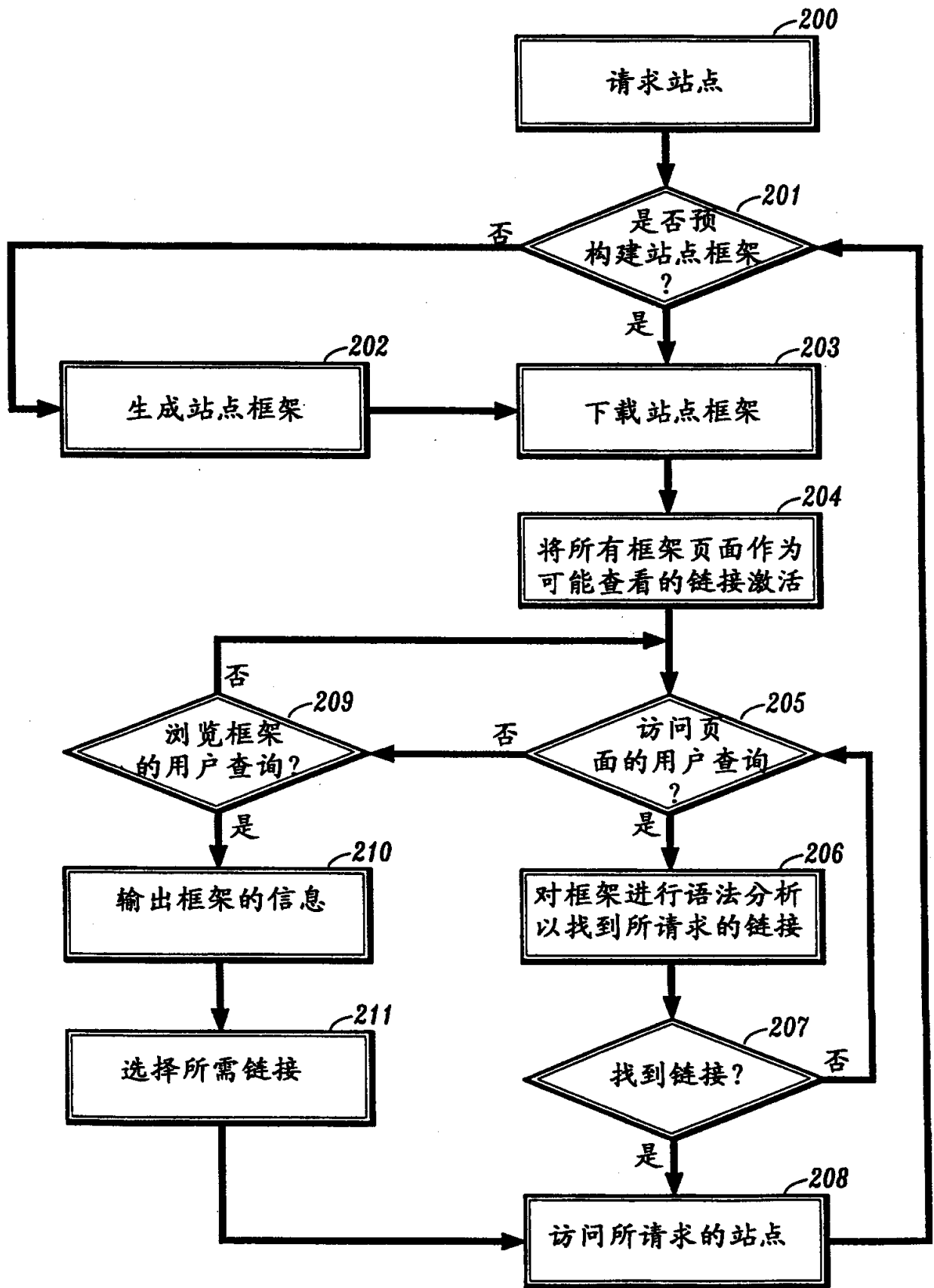


图 2