



# [12] 发明专利说明书

[21] ZL 专利号 97114698.5

[43] 授权公告日 2003 年 5 月 21 日

[11] 授权公告号 CN 1109306C

[22] 申请日 1997.7.18 [21] 申请号 97114698.5  
 [30] 优先权  
 [32] 1996. 8. 19 [33] DE [31] 96113274.3  
 [71] 专利权人 国际商业机器公司  
 地址 美国纽约州  
 [72] 发明人 T·彼德罗夫 H·—J 里希施泰因  
 H·维特曼  
 [56] 参考文献  
 US5491820 1996.02.13 G06F13/00  
 US5530821 1996.06.25 G06F12/10  
 审查员 林松岭

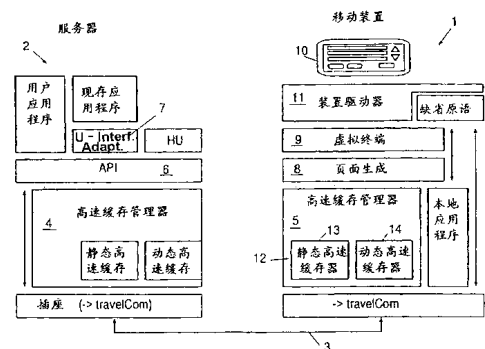
[74] 专利代理机构 中国专利代理(香港)有限公司  
 代理人 王 勇 陈景峻

权利要求书 3 页 说明书 23 页 附图 3 页

[54] 发明名称 在用户机 - 服务器对话系统中执行交互应用的方法、系统及装置

### [57] 摘要

揭示了一种在基于客户机-服务器的交互式处理系统中执行交互式应用程序的机制，其中这些应用程序在客户机中以文件表示。建议独立于为表示各个文件页面所需的客户机系统资源来说明文件，和仅确定用于所建立页面的基本功能特征。为了判断用于表示应用程序的所有可能的系统资源，会话的说明被限制于仅仅为功能特点，象为运行应用程序所需的最小必须的用户交互作用装置。



1. 基于客户机 - 服务器的对话系统中执行交互式应用程序的方法，所述应用程序采用文件 (20-23) 与客户机 (1) 进行交互操作，和所述客户机提供为运行所述应用程序和为表示所述文件的系统资源 (10, 40, 41)，所述方法的特征在于下列步骤：
- 5        在服务器侧 (2) 就所建立页面和就页面信息内容说明所述文件，和生成独立于所述系统资源的文件信息；
- 将所述文件信息由服务器传送 (3) 到客户机；
- 就所述被传送的文件信息在客户机侧生成 (8) 文件，所述文件独立于所述系统资源；
- 10       在客户机侧就所述系统资源生成 (11) 文件页面；和
- 在客户机侧表示 (10) 所述文件页面。
2. 按照权利要求 1 的方法，其特征是就各个应用程序的功能需要将被建立的所述文件页面映射到客户机的系统资源。
- 15       3. 按照权利要求 1 的方法，其特征是所述建立页面的信息由功能对象和表现对象表示。
4. 按照权利要求 1 的方法，其特征是按照客户机所表示的文件页面的实际系统资源需要，提供用于自动调整和配置客户机的系统资源的系统资源调整措施。
- 20       5. 按照权利要求 3 的方法，其特征是将涉及有关文件的交互性需要的信息连同所说的表现对象一齐传送。
6. 按照权利要求 1 的方法，其特征是在客户机和/或服务器侧高速缓存 (4, 5, 12~14) 文件。
7. 按照权利要求 1 的方法，其特征是在传输前压缩信息流和在传输后扩展该信息流。
- 25       8. 按照权利要求 1 的方法，其特征是在服务器与客户机之间双向传送用户操作信息。
9. 按照权利要求 1 的方法，其特征是由服务器向客户机传送为用户在客户机侧指定可能的交互操作所需的会话或交互作用语句。
- 30       10. 按照权利要求 1 的方法，其特征是以至少一个表示多媒体表示对象的可扩展逻辑框 (20~23) 说明文件，和表示所述可扩展逻辑框内的所述文件页面信息内容。

11. 按照权利要求 10 的方法, 其特征是以多个连续行 (21、22) 说明所述文件, 其中每一行至少由一可扩展逻辑框构成。

12. 按照权利要求 1 的方法, 其特征是提供在激活后自动由客户机向服务器自动传送的自动发送对象。

5 13. 为执行提供与客户机 (1) 作交互操作的文件的交互式应用程序的客户机-服务器对话系统, 其中所述客户机提供为运行所述应用程序和为表示所述文件所需的系统资源 (10), 其特征是所述对话系统包括:

10 在服务器侧 (2) 为生成关于建立的页面和关于独立于所述系统资源的页面信息内容的文件信息的装置 (6);

为由服务器至客户机传送所述文件信息的装置 (3);

客户机侧 (1) 的页面生成装置 (8), 用于就所述页面建立信息和就所述被传送的文件信息生成文件页面;

15 客户机侧的资源管理装置 (11), 用于按照所述被传送的文件信息使所述文件页面的页面建立与所述系统资源适配; 和

在客户机侧为表示所述文件页面的装置 (10)。

14. 按照权利要求 13 的对话系统, 其特征是包括有:

静态高速缓存装置 (13), 用于高速缓存在应用程序的运行期间预装载和不可改变的文件信息;

20 动态高速缓存装置 (14), 用于高速缓存在所述运行时间的期间内采集的信息; 和

高速缓存管理装置 (4, 5), 用于控制所述静态和动态高速缓存装置。

25 15. 按照权利要求 13 的对话系统, 其特征是还包括有虚拟终端装置 (9), 用于在所述系统资源上映射文件页面, 所述文件页面由所述页面生成装置 (8) 所生成。

16. 按照权利要求 13 的对话系统, 其特征是还包括压缩和扩展所述被传送的信息的装置。

30 17. 按照权利要求 13 的对话系统, 其特征是客户机还包括有为所述会话表示抽象的多媒体功能单元的代表装置。

18. 按照权利要求 13 的对话系统, 其特征是包括系统资源适配装置, 用于按照用户所表示的文件页面的实际系统资源需要自动调整和

配置客户机的系统资源。

19. 在基于文件(20-23)的客户机-服务器对话系统中用于表示交互式应用程序的终端装置, 其中一文件的表示独立于所述终端装置的各自的系统资源(10), 其特征是所述终端装置包括有:

- 5 页面生成装置(8), 用于按照接收的文件信息生成文件页面;  
虚拟终端装置(9), 用于提供包括由各个应用程序说明的所建立的页面的虚拟文件页面信息;

资源管理装置(11), 用于生成有关所述被提供的虚拟文件页面信息和所述被接收的文件信息的文件页面。

- 10 20. 按照权利要求19的终端装置, 其特征是所述虚拟终端装置将所述虚拟文件页面信息映射到所述系统资源。

21. 按照权利要求19的终端装置, 其特征是还包括有:

静态高速缓存装置(13), 用于高速缓存在应用程序运行时间的期限内预装载和不可改变的文件信息;

- 15 动态高速缓存装置(14), 用于高速缓存在所述运行时间期限内采集的信息; 和

高速缓存管理装置(4、5), 用于控制所述静态和动态高速缓存装置。

20

在用户机--服务器对话系统中  
执行交互应用的方法、系统及装置

<sup>5</sup> 技术领域:

本发明涉及到以客户机-服务器为基础的交互式处理系统中执行的交互式应用程序,其中这些应用程序以使得能与用户对话的文件方式存在于客户机中。具体说,本发明是关于在一移动的客户机通过无线网络与应用程序服务器进行通讯时的移动计算。

背景技术: 在利用象手持式、膝上式(笔记本式)、掌式、或如汽车中的车载计算机之类的便携式计算机的移动网络计算场合,在便携式终端装置(客户机)与固定的应用服务器之间通过无线通信网络保持通信连接。从而就涉及到下列的边缘条件:

1. 共同的客户机硬件的性能不超过磁盘操作系统(DOS系统)的性能,因此与所提供的系统资源相比而言应用软件必须紧凑。此外,为与服务器进行通信需要有通信平台,这时无线电网络的整体性能也带来影响;

2. 与台式计算机或工作站相比当前所采用的移动式计算机的显示器资源变动很大,而且由于该领域的现有技术的缘故在这上面的资源相当有限。此外这些显示器经常是相互间具有十分不一致的显示格式;

3. 当前采用的移动计算机为用户输入/输出信息仅提供有限的并有很大变化的资源。例如,汽车工业中所采用的车载计算机仅具有少量的易于操作的输入键。另一方面,其他装置则设置完全键盘。任何不足额均由安装在客户机侧对应用程序透明的驱动器软件所作的全值输入装置模拟进行补偿;

4. 因而,被用于移动计算的网路成本上很浪费,并且仅能提供低传送率。

在一主机计算机、如IBM系统/370与外设个人计算机(PC)间的对话通信方面存在有一般的途径。安装在各客户机中的专用驱动器软件使得能够实现3270终端模拟。这些系统中提供的应用程序均不了解客户机能与一或多个用户作交互对话的能力,而且这些客户机的显示

资源被分类成有限组的终端类别。在这些显示类别之一的最小显示尺寸为每行每列 80×24 字符。所提供的应用程序用于确定客户机侧的对话页面的格式，从而它们必须了解系统资源以便在客户机侧描述应用程序和在需要时分别配置该应用程序。因此这种方法仅仅只能适用于  
5 与各客户机侧所提供的各个硬件平台无关的统一的主机通信（会话）。

另一通用方面是“超文本传送协议（HTTP）”通信协议，例如在互连网络（Internet）中所普遍应用的。这一协议已使得能够在客户机侧利用不同的硬件平台，由此仅采用一个专用协议。但是，首先此互连网是一种基于信息的系统而不是基于对话的系统，它只是提供一  
10 能为所谓的“网络浏览器（Web Browser）”浏览的信息存放区。其次，这些浏览器的交互式措施是极其有限的，并且仅仅注重于浏览机制和信息的上、下载。而且，为实现这些浏览所需的系统资源超过了移动计算机系统所能允许的大小。

除上述途径外，当前的移动计算领域中，现有的便携计算机与局  
15 部安装的软件协同工作，从而固有的工作性能上受到限制。特别是应用程序的每一变化均必须满足每一个客户机，这就造成额外的费用和转变期间的失败次数，而且还需要用作临时测试过程的时间。

发明内容：因此，本发明的目的就是提出一种方法和系统，它能免除上述现有方案的缺点和使得可能在一客户机-服务器信息处理系统中实现独  
20 立于装置和优化传输的对话。而且本发明还将使得有可能服务器提供的应用程序的改变或修正只需在服务器侧完成。

本发明提供了基于客户机-服务器的对话系统中执行交互式应用程序的方法，所述应用程序采用文件与客户机进行交互操作，和所述客户机提供为运行所述应用程序和为表示所述文件的系统资源，所述  
25 方法的特征在于下列步骤：

在服务器侧就所建立页面和就页面信息内容说明所述文件，和生成独立于所述系统资源的文件信息；

将所述文件信息由服务器传送到客户机；

就所述被传送的文件信息在客户机侧生成文件，所述文件独立于  
30 所述系统资源；

在客户机侧就所述系统资源生成文件页面；和  
在客户机侧表示所述文件页面。

此外，就各个应用程序的功能需要将被建立的所述文件页面映射到客户机的系统资源。

所述建立页面的信息由功能对象和表现对象表示。

5 按照客户机所表示的文件页面的实际系统资源需要，提供用于自动调整和配置客户机的系统资源的系统资源调整措施。

将涉及有关文件的交互性需要的信息连同所说的表现对象一齐传送。

在客户机和/或服务器侧高速缓存文件。

在传输前压缩信息流和在传输后扩展该信息流。

10 在服务器与客户机之间双向传送用户操作信息。

由服务器向客户机传送为用户在客户机侧指定可能的交互操作所需的会话或交互作用语句。

以至少一个表示多媒体表示对象的可扩展逻辑框说明文件，和表示所述可扩展逻辑框内的所述文件页面信息内容。

15 以多个连续行说明所述文件，其中每一行至少由一可扩展逻辑框构成。

提供在激活后自动由客户机向服务器自动传送的自动发送对象。

20 本发明还提供了为执行提供与客户机作交互操作的文件的交互式应用程序的客户机-服务器对话系统，其中所述客户机提供为运行所述应用程序和为表示所述文件所需的系统资源，其特征是所述对话系统包括：

在服务器侧为生成关于建立的页面和关于独立于所述系统资源的页面信息内容的文件信息的装置；

为由服务器至客户机传送所述文件信息的装置；

25 客户机侧的页面生成装置，用于就所述页面建立信息和就所述被传送的文件信息生成文件页面；

客户机侧的资源管理装置，用于按照所述被传送的文件信息使所述文件页面的页面建立与所述系统资源适配；和

在客户机侧为表示所述文件页面的装置。

30 此外，上述对话系统，包括有：静态高速缓存装置，用于高速缓存在应用程序的运行期间预装载和不可改变的文件信息；

动态高速缓存装置，用于高速缓存在所述运行时间的期间内采集

的信息；和

高速缓存管理装置，用于控制所述静态和动态高速缓存装置。

上述系统还包括有虚拟终端装置，用于在所述系统资源上映射文件页面，所述文件页面由所述页面生成装置所生成。

5 上述系统还包括压缩和扩展所述被传送的信息的装置。

其中客户机还包括有为所述会话表示抽象的多媒体功能单元的代表装置。

上述系统还包括系统资源适配装置，用于按照用户所表示的文件页面的实际系统资源需要自动调整和配置客户机的系统资源。

10 在基于文件的客户机-服务器对话系统中用于表示交互式应用程序的终端装置中，一文件的表示独立于所述终端装置的各自的系统资源，其特征是所述终端装置包括有：

页面生成装置，用于按照接收的文件信息生成文件页面；

15 虚拟终端装置，用于提供包括由各个应用程序说明的所建立的页面的虚拟文件页面信息；

资源管理装置，用于生成有关所述被提供的虚拟文件页面信息和所述被接收的文件信息的文件页面。

所述虚拟终端装置将所述虚拟文件页面信息映射到所述系统资源。

20 上述终端装置还包括有：

静态高速缓存装置，用于高速缓存在应用程序运行时间的期限内预装载和不可改变的文件信息；

动态高速缓存装置，用于高速缓存在所述运行时间期限内采集的信息；和

25 高速缓存管理装置，用于控制所述静态和动态高速缓存装置。

本发明为实现上述目标，在一通用的基于客户机-服务器的信息处理系统中提供一种处理口语式对话的机制，各个的对话系统体系结构，和能被用于客户机侧使得能够进行独立于各个基础客户机硬件的多媒体表述的特定的终端装置。特别是，此客户机终端装置能针对一  
30 具有特定硬件要求的应用程序提供自配置。

本发明的基本观念是说明独立于用于表示各个（文件）页面的客户机系统的交互式对话的文件，和仅仅确定所建立页面的主要功能特



点。为了合理对待所有可能的用于表示应用程序的系统资源，会话的说明（对话和所期望的响应）被仅限于如为运行一定的应用程序所需的最少的必要用户交互式装置（按钮等）这样的功能方面。

5 所建立的实际表示由被置于客户机侧的硬件专用资源管理装置产生，在此确定由用户例如通过键盘、触摸屏幕、或软件模拟按钮进行的交互作用的可能模式。这一装置还能用于表示客户机侧的抽象的功能性多媒体单元。在一优选实施例中，此管理装置被作成为一置于网络与客户机表示装置（如显示器）之间的资源适配软件层。应用软件只需利用可由各个现有客户机实现的多个反馈机构。

10 在服务器与客户机间传送有关对话的信息之前，文件描述是通过分离仅用于在客户机侧建立页面而不用于实际用户会话的页面建立信息，和有关用户与应用程序会话的并在会话后阶段加以传送的页面信息内容进行的。此页面建立信息可在开始实际会话之前，例如依靠将该信息在客户机侧作（予）高速存贮被设置在客户机侧。

15 因此，本发明使得一服务器系统所设定的应用程序能在包括有各种不同的用于表示的系统资源的远地客户机处作会话表示。应用程序的进一步改变或一应用程序中的修正只须在服务器侧向不必对所有的客户机进行。

而且上述基本通信网络的缺点也在所推荐的高速缓存机制（智能高速缓存）和保证将被传送的信息降低到最小从而增强了移动计算的性能和可租用性的高速协议中给予了考虑。在此，文件的部分内容被快速存到客户机侧。为此最好设置一静态高速缓存用于高速存放被预先装载并在运行一应用程序时不改变的文件信息，和一动态高速缓存用于高速存放在应用程序运行期间收集的信息，此二高速存贮器可由一高速存贮管理器控制。

25 对于面向对象技术而言，页面建立信息最好能以功能对象和表示对象来表达，其中功能对象可独立于客户机表示硬件平台加以定义。由于这种推荐的结构，文件对一特定客户机硬件的适配可自动地完成并且对用户是透明的。

30 在本发明的另一实施例中，提供涉及到例如声音、按钮、屏幕建立等的文件的多媒体表示的特殊对象。还可提供抽象的多媒体功能单元来使得能实现自动适配原先不知道的客户机多媒体资源的多媒体表

示。

文件页面的表示并不局限于在一计算机显示器上的可视页面。而且在实现多媒体成份如声音、动画等上面也无任何限制。

被传送的信息流为使传送所需带宽最小还可加以压缩和扩展。为了能传送服务器侧为运行如应客户器侧的用户查询而作的服务器访问这

样的应用程序所需的特定信息，最好能在服务器与客户器之间提供双向信息流。例如，客户机可通知服务器关于它的用于表示多媒体表示对象的系统资源。

5 所推荐的机制和结构最好能利用独立于客户机的各自的显示格式并能自动适应这些格式的可扩展的逻辑框来实现，以此来维持逻辑相互间的相对尺寸。各逻辑框可包含一多媒体表现对象，为此一可扩展逻辑框的涵义取决于各个服务器并由各个应用程序来确定。在本发明的一优选例中，这些逻辑框按在一文件中逐行设置。

10 客户机侧提供的对话掩模常行包含输入阵列，这里与整个文件由客户机至服务器的中介传输不同，仅须将对话掩模的一小部分加以更新。例如在图 3 所示的医学应用中，敲击阵列中的按钮区 30 将使得显现一系列健康保险公司，这时可选择这些公司中之一来用于对话的下一步骤。这一点是由利用无需用户参与而由客户机发送往服务器的所谓的“自动发送字段”来完成的。

15 <sup>附图说明：</sup>参照所列附图将会进一步理解本发明的这些和其他的目的、特色和优点。所列附图为：

图 1 为说明按照本发明一优选结构的方框图，其中一移动客户机被连接到一固定的应用程序服务器；

图 2 表明根据可扩展逻辑框建立的示例的可视（文件）页面；

20 图 3 表明医学病员信息处理领域中的示例对话；和

图 4a 和 4b 为具有表示各自的对话掩模的不同系统资源的二个客户机所表明的同一应用程序对话的二个瞬象。

<sup>具体实施方式：</sup>客户机 - 服务器结构

25 图 1 以方框图表明一客户机 - 服务器系统的优选体系结构，其中移动客户机 1 与一可以是任何适用于计算机系统的服务器的固定的服务器 2 进行通信。另外不假定此服务器与客户机通过一无线网络 3 相互连接。这里应指出的是本发明并不局限于无线电网络，而可以是象综合业务数字网络（ISDN）或其他广域网（WAN）那样的远程通信网络，或者如互连网（Internet）或专用内部网络（Intranet）之类的局域网（LAN），或者甚至象大部分公共商店中所设置的红外（IR）线控制的通信网络。

30

运行应用程序涉及到有关应用程序在客户机侧的表示，特别是其对

话式的表示方面的服务器与客户机之间的信息交换。这种信息可被划分为与建立表示有关的信息和可被看作为文件的信息内容的、由用户交互作用所交换的信息。这些信息流通过被设置在服务器 2 和客户机侧 1 的高速存贮器管理模块 4、5。对于任何作高速缓存的信息，仅需传送一基准信息。无须穿过高速存贮管理模块 4、5 的仅有的例外信息是在服务器 2 与客户机 1 间进行交换的基本控制和命令（查询）信息。

另外还设置一应用程序接口（API）6，依靠它可以在服务器上运行所有种类的应用程序，或者直接利用 API，或者依靠插入一例如可被用于主机-PC 通信的适配层 7。

10 所传送的信息流最好相对于已在客户机侧 1 作高速缓存的信息加以简化。信息流通过无线电传输线 3 被传送到移动客户机 1。客户机侧 1 的高速缓存管理模块 5 将信息流重新扩展到原先的形状。对于文件内容信息，被扩展的信息流由一页面生成模块 8 解释并被送往虚拟终端模块 9。此虚拟终端模块根据页面的抽象和功能性说明产生实（文件）页面，  
15 依靠取决于硬件的装置驱动器单元 11 在现有的客户器硬件 10 上进行表示。

在这一实施例中，此装置驱动器单元 11 另外还提供多个表示原语，藉此可由客户器终端表示象警告信号、遥测数据等的独立于虚拟终端模块 9 进行表示的附加通告。由此而能处理客户机所提出的可行的局部应用程序，例如为查询象汽车的发动机数据或全球定位系统（GPS）数据等的遥测数据的应用程序。这些数据也可依靠取决于装置的驱动器单元 11 的表示原语来表示。

#### 文件说明

25 按照本发明，文件的说明、亦即建立和信息内容是抽象的，和对于客户机的各个表示资源为装置独立的。文件的具体的实际意义首先是在客户机的终端装置 10 上例如作为页面表示时提供的。表示的外观可能随各个客户机的表示能力改变。但是由于抽象的说明文件的基本单元以及因而功能性仍能保存住。

30 按照所建议的机制存在有生成文件的多个实施方案。在第一实施方案中，借助形式描述语言可完成纯粹文本形式的文件生成。在另一实施方案中借助各自的 API 功能以页面单元的结构来实现。由此用于内部再现文件所需的各个数据结构明确地产生在应用程序的程序代码之内，而

说明则由形式语言通过语法分析程序变换成内部页面表示。

上述实施方案中，文件被再分成一静态部分，它相当于用于控制对应于应用程序与用户之间的会话对答的字符模式的部分的保留或删除的对话掩模；和一第二动态部分，它相当于文件的输入和输出区的内容或用户所作的实际选择区域。在传送之前，此二部分被置于仅仅一个文件中，但在传送阶段被分开处理。对于已在客户机侧 1 设定（亦即被高速缓存的 12）文件的静态部分，仅将符合的动态部分传送到被插入在一现有的掩模中的客户机。

在此实施方案中，文件以多个叠加的行来说明。每一行由至少一个边靠边地配置的逻辑框构成，这些逻辑框自动地扩展到所定的表示格式并相当于至少一个象输出信息（如正文）或一输入字段那样的组成单元。一行的高度由所包含的最大逻辑框确定。一行内的逻辑框可以相对于最大行高被置为中心，或者可以位于一行的顶部或底部。

行 20 中表示的逻辑框 20 被作边靠边的安排，但也可扩展到最宽行的最大宽度，不过至少为各个客户机的表示装置的最大实际宽度。这一点参照图 2 加以说明，其中总体宽度由第二行 22 的宽度确定。第一行中表示的逻辑框 20 相应地被扩展到该宽度。依靠在被表示的功能逻辑框之前置入、或附加到其后（空的）可扩展逻辑框 23，就可作为将该功能逻辑框表示在中央或者右边对齐的位置。各 AUTO\_FILL（可扩展的）逻辑框就同一行（行）中现有的 AUTO\_FILL 逻辑框的总数而言，具有相等百分比的左自由空间；从而使得能随意地、但与装置无关地定位逻辑框。

此外，一逻辑框必定不仅仅包含功能单元，因而能含有多行，或另外的逻辑页面。因此能进行任何种类的递归式嵌套。这样就可随意设计文件而不致影响所期望的系统资源的独立性。

按上述的说明语言概念生成的（文件）页面的实际具体大小仅仅只由客户机侧提供的表示装置确定。因此，从应用程序的观点上看，无法考虑表示的绝对尺寸。只有字符宽度或用于输入和输出字段的行数可在各异的界限内加以确定。但最终的大小取决于所采用的字符组。

各可扩展逻辑框均表示所说明文件的一或多个的个别功能单元的帧。会话可被设计为仅只是将这些单元进行组合。上述优选实施方案的符合单元如表 1 中所说明的，这仅仅是所涉及到的能被各个客户机硬件

所支持的所有类别的多媒体成分的示例。由指定“**AUTO\_FILL**”的字段宽，就将各个可扩展逻辑框增宽到使整个行宽度被扩展到最宽行的宽度，而至少达到提供到显示器的实际宽度。如果不只一个字段含有这一参数，所有这些逻辑框均将成比例扩展。输入字段的“自动启动”属性和 **n** 选择字段之 **1** 表明任何选择或输入检验均被作为一按钮的激活加以处理，亦即一符合消息被送到服务器。下面给出为实现这些功能的形式说明语言。

每一字段单元均包含多个属性，其中，这些属性的部分对所有类型字段均相同，而其它的则是对每一字段或多个字段所专用的。对于未确定宽度和/或高度的字段，各自的逻辑框的大小根据所表示的字段内容进行计算。在一字段预定有宽度和/或高度时，例如象所选字符组的标准字符行的倍数，则将一最后宽阔的表示，随字段类型而定，被分割、分解、或人工上卷成为表示空间的可视部分。

图 4a 和 4b 中表明二不同的客户器终端装置 40、41 上的应用程序对话掩模的显示器表示，这些客户机具有用于表示文件的不同系统资源。图 4a 中的装置具有一显示区 42，三个作“**OK**”、“**Cancel**”和“**Func**”操作的功能键 43、44、45，以及作显示引导的箭示键 46。在此显示器上表明一应用程序对话框的当前瞬象。

与此相反，图 4b 中说明的终端装置包括有一不同形状和大小的显示器 47，代替功能键 43、44、45 及箭示键 46 的八个功能键（“**F1 - F8**”）48，和一作显示引导的跟踪球 49。

如可由图 4b 看到的，应用程序窗 47 包括一包含多个象监视器型号、主板型号或声卡的型号那样的计算机部件的上面部分 50。在另一窗口部分 51 中，对于上面部分 50 中每一项显示可用的背景信息。如在此显示器的下面部分 52 向用户所通告的，按压“**F1**”按钮可选择一项，或者按压“**F2**”按钮取消选择该项。用户可利用跟踪球 49 滚动通过上部显示器窗，此时有效项由连接到滚动条（滑标）53 的光标条增亮。应予强调的是，在图 4b 的终端装置 41 的情况下，所需的交互式措施（“**OK**”和“**Cancel**”功能）被映射到实际的现有功能键 48，此时这种映射也在屏幕 47 上在位于显示器 47 的底部的附加信息行 52 中加以显示。

图 4a 中所示的装置 40 的显示区 42 与图 4b 中所示的显示器 47 相

比较特别具有较小的尺寸。结果，在显示器 42 的右侧显示另一滑标 54，因为当前的页面需要有较之客户机终端 40 实际提供的更多的垂直空间。

#### 页面说明语言

- 5 下面说明能实现以上功能的形式说明语言。采用这种形式语言（可以更有利地）描述会话式对话和交互式特点。图 3 中的示例对话可按照下面的伪代码来加以说明：

### 10 (\*请求病人数据的对话\*)

```
DOCUMENT REPLIES(OK, CANCEL, F0:"Hilfe")

ROW (*对话标题*)

    STATIC FONT_SIZE(LARGE) FONT_COLOR(INTENSE)
        H_ALIGN(CENTER) WIDTH(AUTO_FILL)
        TEXT("Patientendaten")

ROW (*病人姓名*)

    STATIC TEXT("Name:")
        INPUT IDENTIFICATION("name") WIDTH(AUTO_FILL)

ROW (*地址/街区*)

    STATIC TEXT("Strasse:")
        INPUT IDENTIFICATION("strasse") WIDTH(AUTO_FILL)

ROW (*地址/地区*)

    STATIC TEXT("PLZ:")
        INPUT IDENTIFICATION("plz") WIDTH(5) INPUT_TYPE(NUMERIC)
        MAX_INPUT(5)
    STATIC TEXT("Ort:")
        INPUT IDENTIFICATION("ort") WIDTH(AUTO_FILL) HEIGHT(2)

ROW (*医疗保险*)

    RADIO POPUP IDENTIFICATION("kk") TEXT("Krankenkasse:")
        LIST_ITEM("AOK":S, "Techniker", "Vereinigte")
```

此示例文件被配置成用于返回用户的标准的应答“OK”、“CANCEL”和“FO”。客户机终端上的这些应答的表示取决于各自的硬件资源。在客户机的硬件对这些功能提供合适的功能单元时，例如一键盘的符合的功能键，在客户机的显示器上就无需表示附加的单元。另一方面，如果客户机不提供这样的键，就可在客户机的显示器上表示各自的功能。

下面表明为说明一文件的形式语言。这里提及到基本数据类型仅仅是为了清楚说明。

10



```

document = "DOCUMENT" [reply] {row}
reply    = "REPLIES" "(" stdButton[:string] {", " stdButton[:string]} ")"
stdButton = ("OK"|"CANCEL"|"YES"|"NO"|"PREV"|"NEXT"|"F"("0"-"9"))
row      = "ROW" {box}
box      = {vAlign} fieldDescr
fieldDescr = (staticOutputField | dynamicOutputField | inputField | radioChoiceField | multiChoiceField | buttonField)
staticOutputField = "STATIC" [fontSize] [fontStyle] [fontColor] [width] [height] [hAlign] [text] [noWordWrap]
dynamicOutputField = "DYNAMIC" [fontSize] [fontStyle] [fontColor] [width] [height] [hAlign] [text] [noWordWrap] [identification]
inputField = "INPUT" [fontSize] [fontStyle] [fontColor] [width] [height] [hAlign] [text] [noWordWrap] [maxInput] [identification] [typeList] [autoActivate]
radioChoiceField = "RADIO" [fontSize] [fontStyle] [fontColor] [width] [height] [hAlign] [text] [identification] [itemList] [autoActivate] [popup]
multiChoiceField = "MULTI" [fontSize] [fontStyle] [fontColor] [width] [height] [hAlign] [text] [identification] [itemList]
buttonField = "BUTTON" [fontSize] [fontStyle] [fontColor] [width] [height] [hAlign] [text] [identification]
vAlign      = "V_ALIGN" "(" ("TOP"|"CENTER"|"BOTTOM") ")"
hAlign      = "H_ALIGN" "(" ("LEFT"|"CENTER"|"RIGHT") ")"
fontSize    = "FONT_SIZE" "(" ("LARGE"|"MEDIUM"|"SMALL") ")"
fontStyle    = "FONT_STYLE" "(" ("NORMAL"|"EMPHASIS"|"ATTENTION") ")"
fontColor    = "FONT_COLOR" "(" ("NORMAL"|"INTENSE"|"LIGHT") ")"
width       = "WIDTH" "(" integer | "AUTO_FILL" ")"
height      = "HEIGHT" "(" integer ")"
text        = "TEXT" "(" string ")"
string      = "" 任意, 除引号外为 ASCII 串 ""
noWordWrap  = "NO_WORD_WRAP"
maxInput    = "MAX_INPUT" "(" integer ")"
identification = "IDENTIFICATION" "(" string ")"
autoActivate = "AUTO_ACTIVATE"
popup       = "POPUP"
typeList    = "INPUT_TYPE" "(" type {", " type} ")"
type        = ("ALPHA"|"NUMERIC"|"CAPITALS"|"INVISIBLE"|"DATE"|"PROTECTED")
itemList    = "LIST_ITEM" "(" string[:S] {", " string[:S]} ")"

```

### 应用程序接口

5 为了为一应用程序生成文件，最好能应用下面的应用程序接口（API）。至于以上讨论的限制，它只包括少量的足以生成所有所需结构的基本功能。

10 文件的内部表示保持对用户不透明。文件存放所传送的信息流（对话掩模）的静态部分，以及用作为预定的正文或用作为对输入字段、对定义动态输出字段、或者对选择字段中的用户选择的解释性注解的动态变化信息。这些动态数据仅存在于欲被传送的信息流的生成阶段，此后将被消除。这里要避免的是，在附加其他动态信息之后，先前的数据被再次传送。依靠利用 API，可在任何时刻修正文件的动态部分并与各自的静态部分（对话掩模）无关地被传送至客户机。

被传送的信息流可被控制命令和查询命令或另外的动态数据加以扩大，而完成的信息包可被“整体地”传送给客户机。

15 以下的以 C 语言编写的程序代码的摘录描述图 3 中所示示例会话（对话）的生成。各个属性各由以前缀“TT\_”扩充的允许说明语言参数组成。

```

TTDocHandle doc;

doc = ttNewDocument();

ttAddReply(doc, TT_OK, NULL);
ttAddReply(doc, TT_CANCEL, NULL);
ttAddReply(doc, TT_F0, "Hilfe");

ttAddRow(doc);

ttAddBox(doc, TT_STATIC, TT_DEFAULT);
ttSetAttribute(doc, TT_FONT_SIZE, (AttribValue) TT_LARGE);
ttSetAttribute(doc, TT_FONT_COLOR, (AttribValue) TT_INTENSE);
ttSetAttribute(doc, TT_H_ALIGN, (AttribValue) TT_CENTER);
ttSetAttribute(doc, TT_WIDTH, (AttribValue) TT_AUTO_FILL );
ttSetAttribute(doc, TT_TEXT, (AttribValue) "Patientendaten");

ttAddRow(doc);

ttAddBox(doc, TT_STATIC, TT_DEFAULT);
ttSetAttribute(doc, TT_TEXT, (AttribValue) "Name:");
ttAddBox(doc, TT_INPUT, TT_DEFAULT);
ttSetAttribute(doc, TT_IDENTIFICATION, (AttribValue) "name");
ttSetAttribute(doc, TT_WIDTH, (AttribValue) TT_AUTO_FILL );

ttAddRow(doc);

ttAddBox(doc, TT_STATIC, TT_DEFAULT);
ttSetAttribute(doc, TT_TEXT, (AttribValue) "Strasse:");
ttAddBox(doc, TT_INPUT, TT_DEFAULT);
ttSetAttribute(doc, TT_IDENTIFICATION, (AttribValue) "strasse");
ttSetAttribute(doc, TT_WIDTH, (AttribValue) TT_AUTO_FILL );

ttAddRow(doc);

ttAddBox(doc, TT_STATIC, TT_DEFAULT);
ttSetAttribute(doc, TT_TEXT, (AttribValue) "PLZ:");
ttAddBox(doc, TT_INPUT, TT_DEFAULT);
ttSetAttribute(doc, TT_IDENTIFICATION, (AttribValue) "plz");
ttSetAttribute(doc, TT_WIDTH, (AttribValue) 5 );
ttSetAttribute(doc, TT_INPUT_TYPE, (AttribValue) TT_NUMERIC);
ttSetAttribute(doc, TT_MAX_INPUT, 5);
ttAddBox(doc, TT_STATIC, TT_DEFAULT);
ttSetAttribute(doc, TT_TEXT, (AttribValue) "Ort:");
ttAddBox(doc, TT_INPUT, TT_DEFAULT);
ttSetAttribute(doc, TT_IDENTIFICATION, (AttribValue) "ort");
ttSetAttribute(doc, TT_WIDTH, (AttribValue) TT_AUTO_FILL );
ttSetAttribute(doc, TT_HEIGHT, (AttribValue) 2 );

ttAddRow(doc);

ttAddBox(doc, TT_RADIO, TT_DEFAULT);
ttSetAttribute(doc, TT_IDENTIFICATION, (AttribValue) "kk" );
ttSetAttribute(doc, TT_POPUP, NULL );
ttSetAttribute(doc, TT_TEXT, (AttribValue) "Krankenkasse:");

ttSetAttribute(doc, TT_LIST_ITEM, (AttribValue) "AOK" );
ttSetAttribute(doc, TT_LIST_ITEM, (AttribValue) "Techniker" );
ttSetAttribute(doc, TT_LIST_ITEM, (AttribValue) "Vereinigte");
ttSetAttribute(doc, TT_SELECT_ITEM, (AttribValue) 0);

```

下面列举由 API 实现的函数。可能的返回码可由指定 API 标题文件的“`tt_api.h`”的各个注解采集。

**TTDocHandle ttNewDocument ( void )**

5 此功能用于产生新的空文件。所返回的句柄必须对该文件的所有操作加以命名。（与“指针”相反，“句柄”是对一对象的抽象的引用（条件））。

**int ttFreeDocument (TTDocHandle handle)**

对每一文件均将内部地分配一些资源号。在该文件不再需要时，这些资源可再次被这一函数完全释放。

10 **int ttFreeDynamic (TT Doc Handle handle)**

这一函数释放一文件的所有动态数据，并将在当文件的动态部分已被函数“`ttAddDyn2DataStream`”插入在一信息流中时自动地执行。

**TTDocHandle ttParseDocument(char\*string)**

15 当 - TTDL 文件被设置在一零终止缓存器中时（C-string），即可由此函数将其变换成一文件句柄。在成功的变换之后，此函数即递交一文件句柄。

**char \*ttLastParseError(void)**

此函数将一指针传递给一指定在扫描、或分析文件期间的各自的最后错误的字符串。

20 **TTDocHandle ttReadDocument(char\*filename)**

在一文本文件提供一 TTDL 格式时，这函数可将其变换成内部格式。在成功的读和变换之后，文件句柄被返回。

**int ttWriteDocument (TTDocHandle handle ,char \*filename)**

25 通过这一函数可将文件句柄指定的文件写入一文件，例如 TTDL 文本文件。

**int ttAddReply (TTDocHandle handle,int button ,char\*modifier)**

由此可对一文件确定一标准应答按钮。如果无须任何经修改的标题，就必须命名各自的“空”指针。

**int ttAddRow (TTDoc Handle handle)**

30 此函数将一新的空行附加到一文件。

**int ttAddBox (TTDoc Handle handle,int fieldType,int V\_align)**

此函数将一新的右侧对齐的逻辑框插入文件的有效行。如果不希望

对逻辑框作任何特定的垂直调整，作为缺省值必须命名“TT\_DEFAULT”。而且必须说明所含功能单元，即“TT\_STATIC”、“TT\_DYNAMIC”、“TT\_INPUT”、“TT\_RADIO”、“TT\_MULTI”、或“TT\_BUTTON”的型式。

5        **int ttSetAttribute(TTDocHandle handle,int attrib\_Type,Attrib Value Value)**

由此函数可确定一有效的、最后生成的、或以“tt Set Current Box”设定的逻辑框的功能单元的所有属性。

**int ttSetCurrentBox(TTDocHandle handle,char'id)**

10        在此，带有识别符“id”的逻辑框可被置为有效，并因而使另外的操作象“tt SetAttribute”涉及到该逻辑框。这一功能可仅仅应用于在逻辑框的组述阶段已对之定义各自的识别属性的逻辑框。

**TTDataStreamHandle ttOpenDataStream (Vaid)**

15        由此函数可建立按照本发明的数据流。此后这一信息流可以任意的文件或动态控制和查询数据加以填充，再进行传送。

**void ttCloseDataStream (TTDataStreamHandle ttds)**

将其句柄连同相关的资源传送到这一函数，即可释放不再需要的数据流。

20        **int    ttAddDoc2DataStream(TTDataStreamHandle    ttds,TTDoc Handle doc)**

在此被传递的文件的静态和动态部分可被插入一给定的数据流。此后舍弃文件的动态信息。

**int    ttAddDyn2DataStream (TTDataStreamHandle    ttds,TTDoc Handle doc)**

25        由这一函数可将所传递的文件的动态部分插入给定的数据流。此后如前面那样舍弃动态信息。在某文件的动态数据有所改变且该文件将被独立地传送到客户机以便将它们插入一有效文件中的情况下，这些动态数据甚至将为数据流所具有的仅有部分。

表 2 中较详细地说明关于函数“ttSetAttribute”的可能的参数。

30        **数据流的编码**

在生成信息流时数据被编码以获得尽可能紧凑的传输信息包。所有字长大于 1 字节的数据均以“网络字节顺序”加以缓存，亦即，字的最

高位字节（MSB）被置于低地址。对于为正整数，采用数据型式“组合的无符号整数”，亦即一字节的7比特（位）被用于表示数的值而MSB被置位表示后随有另外的字节。下面以与前面所示代码示例相同的符号说明这样的数据流的结构。任何注解均置入括号中并以草写体表示在各行的相应尾端。

packet = counter {(docData | dynamicData | control)}  
 counter = UCHAR (incremental counter of sequential packets MOD 256)

docData = docTag docId length cacheCtrl [reply {row}+] (对于 Cache Ctrl>0  
 无回答和行)  
 docTag = UCHAR (TTDS\_DOC\_DATA)  
 reply = length {replyButton}  
 replyButton = (buttonId | modifiedButtonId string)  
 modifiedButtonId = buttonId, logically or'ed with TTDS\_BUTTON\_MODIFIED  
 row = length {box}+  
 box = length fieldTag {attribute}

fieldTag = UCHAR (TTDS\_STATIC, TTDS\_DYNAMIC, TTDS\_MULTI,  
 TTDS\_RADIO, TTDS\_BUTTON)

attribute = (stringAttrib string | enumAttrib enumVal | dimAttrib dimension | inputTypeAttrib inputType | flagAttrib)

stringAttrib = UCHAR (TTDS\_TEXT, TTDS\_LIST\_ITEM)

enumAttrib = UCHAR (TTDS\_BOX\_V\_ALIGN, TTDS\_H\_ALIGN,  
 TTDS\_FONT\_SIZE, TTDS\_FONT\_STYLE, TTDS\_FONT\_COLOR)

enumVal = UCHAR (TTDS\_BOTTOM, TTDS\_CENTER, TTDS\_TOP,  
 TTDS\_LEFT, TTDS\_RIGHT, TTDS\_NORMAL, TTDS\_LARGE,  
 TTDS\_MEDIUM, TTDS\_SMALL, TTDS\_EMPHASIS,  
 TTDS\_ATTENTION, TTDS\_INTENSE, TTDS\_LIGHT)

dimAttrib = UCHAR (TTDS\_WIDTH, TTDS\_HEIGHT, TTDS\_MAX\_INPUT)

inputTypeAttrib = UCHAR (TTDS\_INPUT\_TYPE)

inputType = UCHAR, Bitfield (Bit 0:TTDS\_ALPHA, 1:TTDS\_NUMERIC,  
 2:TTDS\_CAPITALS, 3:TTDS\_INVISIBLE, 4:TTDS\_DATE,  
 5:TTDS\_PROTECTED)

flagAttrib = UCHAR (TTDS\_NO\_WORD\_WRAP, TTDS\_AUTO\_ACTIVATE,  
 TTDS\_POPUP)

dynamicData = dynDataTag docId length cacheCtrl {dataField} (对于 Cache Ctrl>0  
 无数据字段)  
 dynDataTag = UCHAR (TTDS\_DYNAMIC\_DATA)  
 dataField = length fieldAddr fieldValue  
 fieldValue = (stringFieldTag string | selectionFieldTag {selection})  
 stringFieldTag = UCHAR (TTDS\_DYNAMIC, TTDS\_INPUT)  
 selectionFieldTag = UCHAR (TTDS\_RADIO, TTDS\_MULTI)  
 selection = POINT (选定项位置)

**control** = ctrlTag length (logonRequest | logonResponse | logoff | clearDoc | reset |  
 buttonPressed)  
**ctrlTag** = UCHAR (TTDS\_CONTROL\_DATA)  
**logonRequest** = logonReqTag logonData  
**logonReqTag** = UCHAR (TTDS\_LOGON\_REQUEST)  
**logonData** = version hwType logonId sessionId staticCacheId dynCacheSize dynCacheID  
**version** = UCHAR (数据流协议版本)  
**hwType** = UCHAR (硬件类型, 可能值见第 19 页 1,5,2)  
**logonId** = string  
**sessionId** = UCHAR  
**staticCacheId** = crc32 (0 意味“无静态缓存”)  
**dynCacheSize** = PUINT (大小以字节为单位的最大值, 0 意味着“无动态缓存”)  
**dynCacheID** = crc32 (0 意味“使用新的空的动态缓存”)

**logonResponse** = logonResponseTag logonResponseData  
**logonResponseTag** = UCHAR (TTDS\_LOGON\_RESPONSE)  
**logonResponseData** = ULONG (位屏蔽, 用法见 19 页 1, 5, 2)  
**logoff** = UCHAR (TTDS\_LOGOFF)  
**clearDoc** = clearDocTag docId  
**clearDocTag** = UCHAR (TTDS\_CLEAR\_DOCUMENT)  
**reset** = UCHAR (TTDS\_RESET)  
**buttonPressed** = buttonPressedTag docId buttonId  
**buttonPressedTag** = UCHAR (TTDS\_BUTTON\_PRESSED)

**docId** = UWORD  
**buttonId** = UCHAR (TTDS\_OK, TTDS\_CANCEL, TTDS\_YES, TTDS\_NO,  
 TTDS\_PREV, TTDS\_NEXT, TTDS\_F0..9)  
**fieldAddr** = rowAddr boxAddr  
**rowAddr** = PUINT  
**boxAddr** = PUINT  
**cacheCtrl** = PUINT (0: 新数据, >0: 缓存参考标号)  
**length** = PUINT (相应序列长度, 长度一字段大小除外)  
**dimension** = PUINT (字符宽或高的单位, 0 解释为 - 1)  
**string** = length {CHAR}  
**crc32** = ULONG



### 高速缓存机制

已在文件的设计阶段将信息流划分成为静态的和动态部分。这些不同的数据以二个相互独立地被高速缓存的分开的数据块加以处理。这样，例如就可以在客户机侧 1 在静态定（预先装载的）高速缓冲存储器 5 13 中整个地设置对话掩模，由此仅仅各自传送掩模的信息内容。

被传送信息的高速缓存是在第一信息流的传送期间对用户透明地完成的，尽管用户可以通过在数据流的生成阶段期间中选择各个的数据块来影响高速缓存器的性能。

字段类型	属性	
<u>固定输出字段</u> ，用于输出作为对话掩模的部分的不改变正文。	· 输出 - 正文 · 字转换 (真, 假)	· 字符大小 (大, <u>中</u> , 小)
<u>动态输出字段</u> ，用于输出作为所表示的信息部分的可变正文。	· 输出 - 正文 · 字转换 (真, 假)	· 字符型式 ( <u>正常</u> , 加重, 注目)
<u>输入字段</u> ，用于输入字母数字数据，可将输入限于特定的数据型式，或者甚至在输入口令时使之不透明。还可插入预定的正文。	· 型式 ( <u>数字</u> , <u>字母</u> , 大写字母, 不可见的日期, 被保护) · 字转换 ( <u>真</u> , 假) · 自动激活 (真, <u>假</u> ) · 预定确定的正文	· 色彩 (正常, 浓色, 淡色) · 对准 ( <u>左</u> , 中, 右) · 以字符表示的字段宽 (1 ... n) 自动填充 · 以字符 1 ... n 表示的字段高
<u>选择字段</u> (n 之 1)，使得能在多个项目的表中选择一项。	· 标题 · 可选择项目表 · 自动激活 (真, <u>假</u> ) · 弹出 (真, <u>假</u> )	
<u>选择字段</u> (n 之 m)，使得能由一相应的表中选择多个单元。	· 标题 · 可选择项目表	

---

<u>按钮</u> ，用于与应用程序服务器交互作用。	· 标注	
----------------------------	------	--

表 1

属性类型	属性值	有效单元
TT_IDENTIFICATION	至字符串的指针	全部
TT_FONT_SIZE	TT_LARGE, TT_MEDIUM, TT_SMALL	全部
TT_FONT_STYLE	TT_NORMAL, TT_EMPHASIS, TT_ATTENTION	全部
TT_FONT_COLOR	TT_NORMAL, TT_LIGHT, TT_INTENSE	全部
TT_WIDTH	TT_FILL_WIDTH, 1..n	全部
TT_HEIGHT	1...n	全部
TT_TEXT	至字符串的指针	全部
TT_MAX_INPUT	1..n	
TT_NO_WORD_WRAP	未用	静态, 输入, 动态的
TT_POPUP	未用	无线
TT_AUTO_ACTIVATE	未用	无线, 输入
TT_LIST_ITEM	至字符串的指针	无线, 多路
TT_INPUT_TYPE	TT_ALPHA, TT_NUMERIC, TT_CAPITALS, TT_INVISIBLE, TT_DATE, TT_PROTECTED	输入
TT_SELECT_ITEM	0...#items-1	无线
TT_SELECT_ITEM	TT_NO_SELECTION, 0...#items-1	多路

表 2

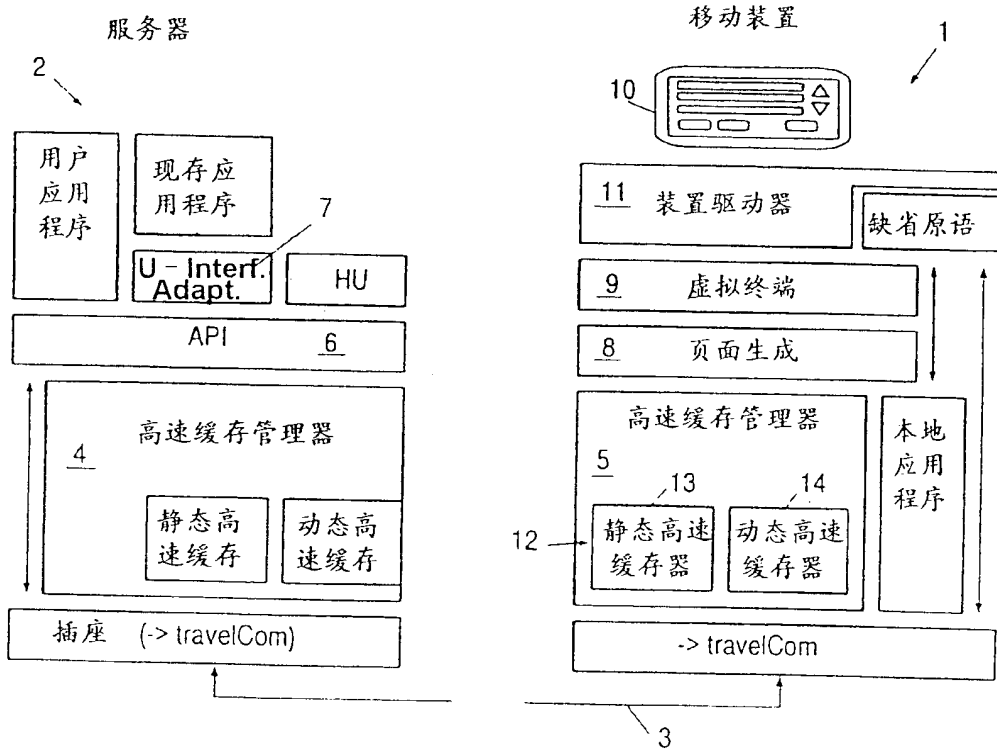


图 1

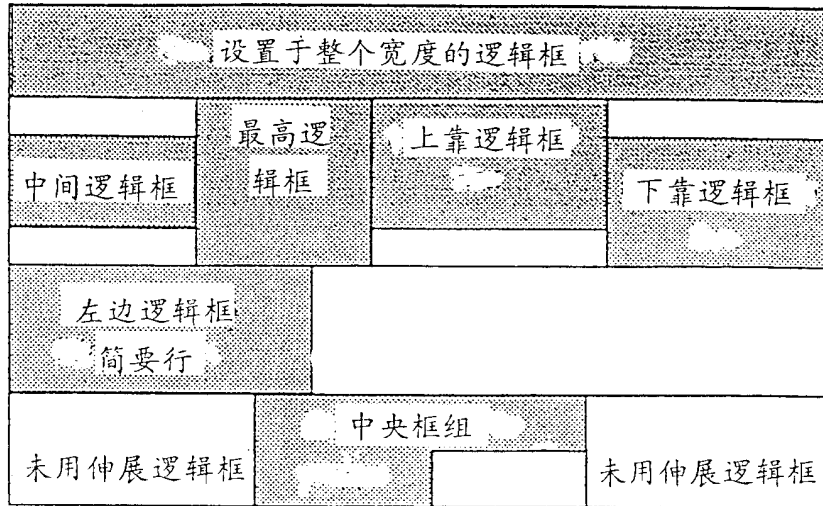


图 2

The form is titled "病人数据" (Patient Data) and contains the following fields:

- 姓名:
- 街区:
- 职务:  地区:
- 疾病保险机关:

A line with the number 30 points to the dropdown arrow.

图 3

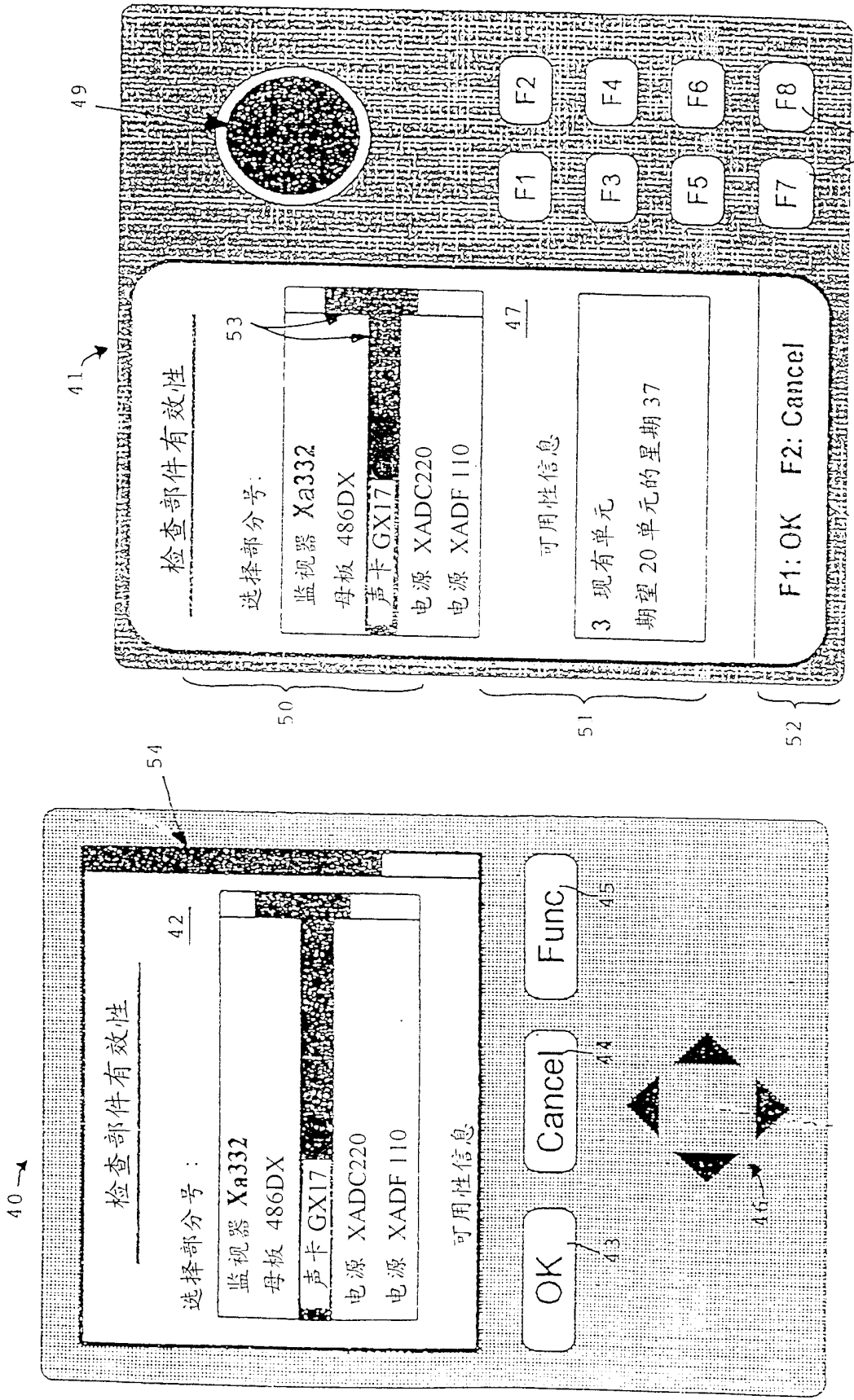


图 4A

图 4B