

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
H04L 12/58 (2006.01)



[12] 发明专利申请公开说明书

[21] 申请号 200510063939.3

[43] 公开日 2006年1月4日

[11] 公开号 CN 1716920A

[22] 申请日 2005.3.30

[21] 申请号 200510063939.3

[30] 优先权

[32] 2004.7.1 [33] GB [31] 0414716.1

[71] 申请人 国际商业机器公司

地址 美国纽约州

[72] 发明人 托马斯·J·G·贝利

克里斯托弗·D·P·詹金斯

乔纳森·M·罗伯茨

基兰·P·斯科特

[74] 专利代理机构 北京市柳沈律师事务所

代理人 邵亚丽 马莹

权利要求书 2 页 说明书 11 页 附图 27 页

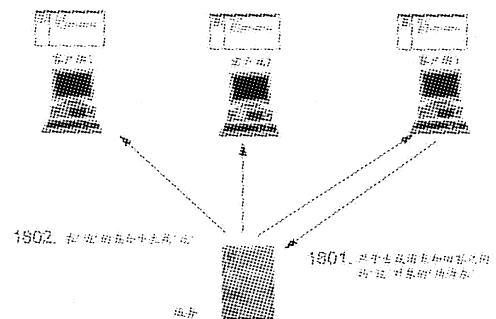
[54] 发明名称

用于即时消息传送系统的会话注释的方法和装置

[57] 摘要

本发明涉及用于即时消息传送系统的会话注释的方法、装置或计算机程序。该即时消息传送方法包含：提供图形用户界面；确定两个或更多个现有消息的选择；根据来自 GUI 的指令，生成所选择的消息之间的关系；和使用消息之间的链接或边，来指示消息之间的关系。所有相关的消息都可以通过单个选择而被选择，并且被成批地打印、存储或删除。

1503 生成和/或接收消息的计算机程序，用于在网络上进行传输。



1. 一种即时消息传送系统中的方法，所述即时消息传送系统具有图形用户界面 GUI，所述方法包含：
- 5 确定即时消息传送系统中的两个或更多个即时消息传送实体的选择；
根据来自 GUI 的指令，生成被选择的即时消息传送实体之间的关系；和
使用 GUI 来指示即时消息传送实体之间的关系。
2. 如权利要求 1 所述的方法，其中，所述即时消息传送实体是现有消息或者消息之间的现有关系。
- 10 3. 如权利要求 1 或 2 所述的方法，其中，所生成的关系是简单链接或者是具有方向的从属链接。
4. 如权利要求 1、2 或 3 中的任何一项所述的方法，还包括：根据通过 GUI 的指令，选择与一个即时消息传送实体有关系的所有即时消息传送实体。
5. 如权利要求 4 所述的方法，还包括：打印、存储或删除所选择的即时
15 消息传送实体。
6. 一种即时消息传送系统，包含：
图形用户界面 GUI；
用于确定即时消息传送系统中的两个或更多个即时消息传送实体的选择的装置；
- 20 用于根据来自 GUI 的指令，生成被选择的即时消息传送实体之间的关系
的装置；和
用于使用 GUI 来指示即时消息传送实体之间的关系装置。
7. 如权利要求 6 所述的即时消息传送系统，其中，所述即时消息传送实体是现有消息或者消息之间的现有关系。
- 25 8. 如权利要求 6 或 7 所述的即时消息传送系统，其中，所生成的关系是简单链接或者具有方向的从属链接。
9. 如权利要求 6、7 或 8 中的任何一项所述的即时消息传送系统，还包括：用于根据通过 GUI 的指令，选择与一个即时消息传送实体有关系的所有即时消息传送实体的装置。
- 30 10. 如权利要求 9 所述的即时消息传送系统，还包括：用于打印的装置，
用于存储的装置，或用于删除所选择的即时消息传送实体的装置。

11. 一种即时消息传送客户端，包含：
图形用户界面 GUI；
用于确定即时消息传送系统中的两个或更多个即时消息传送实体的选择的装置；
- 5 用于根据来自 GUI 的指令，生成被选择的即时消息传送实体之间的关系
的装置；和
用于使用 GUI 来指示即时消息传送实体之间的关系
的装置。
12. 如权利要求 11 所述的即时消息传送客户端，其中，所述即时消息传
送实体是现有消息或者消息之间的现有关系。
- 10 13. 如权利要求 11 或 12 中的任何一项所述的即时消息传送客户端，还
包括：用于根据通过 GUI 的指令，选择与一个即时消息传送实体有关系的所
有即时消息传送实体的装置。
14. 如权利要求 13 所述的即时消息传送客户端，还包括：用于打印的装
置，用于存储的装置，或用于删除所选择的即时消息传送实体的装置。
- 15 15. 一种用于即时消息传送系统的计算机程序产品，所述计算机程序产
品包含存储在计算机可读存储介质上的计算机程序指令，用于当被加载到计
算机中并被执行时，使计算机实施下列步骤：
提供图形用户界面 GUI；
确定即时消息传送系统中的两个或更多个即时消息传送实体的选择；
- 20 根据来自 GUI 的指令，生成被选择的即时消息传送实体之间的关系；和
使用 GUI 来指示即时消息传送实体之间的关系。
16. 如权利要求 15 所述的计算机程序产品，其中，所述即时消息传送实
体是现有消息或者消息之间的现有关系。
17. 如权利要求 15 或 16 所述的计算机程序产品，其中，所生成的关系
- 25 是简单链接或者具有方向的从属链接。
18. 如权利要求 15、16 或 17 中的任何一项所述的计算机程序产品，还
包括：根据通过 GUI 的指令，选择与一个即时消息传送实体有关系的所
有即时消息传送实体。
19. 如权利要求 18 所述的计算机程序产品，还包括：打印、存储或删除
- 30 被选择的即时消息传送实体。

用于即时消息传送系统的会话注释的方法和装置

5 技术领域

本发明涉及用于即时消息传送系统 (instant messaging system) 的会话注释 (conversation annotation) 的方法和装置。

背景技术

10 即时消息传送系统允许在用户计算机之间通过网络进行基于文本的实时会话。这样的系统包含在服务器计算机上运行的服务器应用程序, 而客户端应用程序在用户计算机上运行。客户端应用程序允许被连接的用户使用图形用户界面 (graphical user interface, GUI) 参加彼此的会话或交谈 (chat)。

在已知的即时消息传送系统中, 用户计算机激活客户端应用程序 (客户端); 客户端连接到服务器应用程序 (服务器), 通知它用户是激活的。给客户端展示其他的激活客户端的列表, 并且客户端能够发起与一个或多个被选择的客户端的交谈。交谈由交谈对象所代表, 并被 GUI 可视化。在交谈被建立后, 参与交谈的每一个客户端在 GUI 中显示按时间排序的消息列表。开始时, 消息列表将是空的, 但是交谈中的用户能够使用他们的客户端把消息
15 加入到消息列表中, 被加入的消息将被放在所有客户端上的当前消息列表的末尾。这允许通过陈述、询问和回答来展开会话。

当前的即时消息传送系统提供了按时间排序的线性会话, 即会话内的消息不引用同一会话内的其他消息。当来自两个不同的话题的消息被交织在一起时, 这可能导致会话中的不确定性。如果来自两个不同话题的问题被同时
25 提出, 则对其中一个话题的回答需要明确地引用问题或者话题, 以便避免不确定性。在一群人之间的面对面会话中, 如果一个人被两个不同的人提出问题, 则此人可以简单地通过面对要向其阐述答案的那人, 来避免关于他们正在回答哪个问题的不确定性。这在所给出的回答和被提出的问题之间产生了引用 (reference)。

30 在因特网新闻组消息发布中不存在这个问题, 这是因为在因特网新闻组消息发布中, 一个人在发布消息的同时, 在消息数据库中定义了分层节点。

当一个人在因特网新闻组中发布消息时，一个人既可以回复现有的消息，也可以开始新的思路。在前一种情况下，在问题和回答之间形成了父子关系。这个关系被很多新闻组阅读器程序(例如, Mozilla 邮件、微软 Outlook Express)图形化地显示出来。

5 但是，实时交谈的优点在于它不具有明显的结构，并且消息可以被很快地输入。一旦消息已经被放入实时交谈中，它就被作为下一条消息显示在按时间排序的消息序列中。

例如，在图 1 中，会话在客户端 1、客户端 2 和客户端 3 之间开始，其中问起了两个问题。假设客户端 3 也在会话中，则客户端 3 将需要在回答中引用初始问题。例如，“是的，我准备吃午饭”或者“是的，我愿意在食堂吃”将会避免由简单的“是的”回答引起的不确定性。因为在实时交谈中，消息不保持对彼此的引用，故客户端 3 无法回复“是的”，并指定该回答涉及第一或第二个问题(或话题)。并且，考虑如下情形，即客户端 2 和客户端 3 同时键入它们的回答，然后客户端 3 在客户端 2 之后不久敲击回车键，以至于客户端 3 没有时间反应。

15 拥有如下的即时消息传送系统是有益的，该系统具有实时交谈的速度优势，并具有允许用户利用例如如上所述的解决了不确定性的信息来注释交谈的功能。即时消息传送系统的已知模型不允许发生引用。因此，需要指定这样一种即时消息传送模型的结构，该模型考虑在消息被发送之后的消息之间的引用。

发明内容

根据本发明的第一个方面，提供了一种即时消息传送系统中的方法，所述即时消息传送系统具有图形用户界面，所述方法包含：确定即时消息传送系统中的两个或更多个即时消息传送实体的选择；根据来自 GUI 的指令，生成被选择的即时消息传送实体之间的关系；和使用 GUI 指示即时消息传送实体之间的关系。

即时消息传送实体可以是现有消息或者消息之间的现有关系。

关系可以被定义为简单链接或者具有方向的从属关系。

30 最好包括根据通过 GUI 的指令，选择与一个即时消息传送实体有关系的所有即时消息传送实体。

被选择的即时消息传送实体可以被打印或被保存或被删除。

问题的解决方案是向用户展示用于生成从一条消息到另一条消息的引用的界面。然后，当回答不确定时，例如，在上面的例子中回答“是的”，则用户可以示出该回答所涉及的问题。

- 5 对于此框架结构，要求每一条消息具有交谈内唯一的引用。这可以采取特定的“唯一身份”属性或合起来是唯一的其他属性的组合的形式。模型还指定了每一条消息由一个对象代表。‘交谈消息’是对象，其包含：文本；格式编排；和特定消息的唯一 ID。

- 10 可以生成存在于新框架结构内的新的对象类，以允许用户从一条消息引用另一条消息。把会话和图形数据结构比较，然后可以把会话内的消息比作图形的节点。然后，消息之间的引用是图形中的边；见图 2。因此，如果用户想从一条消息引用另一条消息，则他们可以在这两条消息之间画一条边；见图 3。‘边 (edge)’ 是边的软件对象表示。

- 15 客户端包含用于生成‘边’和用于将‘边’的生成通知所有其他客户端的装置。‘边’包含它引用的每一条消息的 ID，并且每一个客户端使用此信息来显示边。可以用很多方式来显示边；例如，可以画从一条消息到另一条消息的箭头来表示如图 3 中所示的引用。

- 20 当要被连接的两条消息都已经被发送并且被显示在每一个用户的 GUI 上时，用户将希望生成‘边’。然后，希望生成‘边’的用户将能够使用客户端 GUI 来指定要连接的消息。完成它的方式将特定于客户端，但是一种可能的解决方案将是允许用户把一条消息拖到另一条消息上，来指定要通过边连接的消息。然后，客户端将生成‘边’并把‘边’发送到所有其他的客户端。

- 25 在优选的即时消息传送系统中，‘边’仅包含对两条消息的引用。将有可能对该模型进行改造，以使‘边’能够包含对多于两条的消息的引用，例如，如果用户希望用一条‘边’把三条或更多的消息链接在一起。但是，此经改造的系统的任何优点也可以通过生成多条‘边’来达到。例如，如果用户希望链接三条消息，消息 1、消息 2 和消息 3，则经改造的模型可以生成引用消息 1、消息 2 和消息 3 的‘边’。通过生成从消息 1 到消息 2 的‘边’和从消息 2 到消息 3 的‘边’，可以在我们所提出的模型中达到同样效果。在此情形下，消息 1 将通过消息 2 间接地引用消息 3，反之亦然（见图 4 和图 5）。
- 30

在即时消息传送系统的一个实施例中，‘边’将不具有方向。在这种情况下

中，两条消息都互相引用。因为消息具有时间顺序，所以不会由于在消息之间具有简单的‘边’而导致不确定性。往回参考图 3，如果用户 1 的问题和用户 3 的响应之间的‘边’是不定向的或定向的，都不会产生误解。但是，允许‘边’是定向的具有优点。在图 2 的不定向‘边’的情况中，无法指定消息 3 引用消息 1 而又不指定消息 1 引用消息 3。使用具有方向的‘边’给出了差别，在图 5 中，利用指向消息 3 的‘边’，可以将消息 1 定义为引用消息 3，而消息 3 不引用消息 1。

不定向‘边’、定向‘边’和间接引用的组合导致了新的选项。在一个例子中，用户能够选择一条消息（始发消息）或消息的组合，包括：

- 10 1. 所有从始发消息通过定向‘边’引用的消息；
2. 所有通过定向‘边’引用始发消息的消息；
3. 所有从不定向‘边’被链接到始发消息的消息；
4. 所有被引用的消息（上面的 1 和 3）；和
5. 所有的消息（上面的 1、2 和 3）。

15 例如，在图 6 所示的会话中，客户端 3 回答“是的”，然后生成从两个问题到回答的不定向‘边’，指示该“是的”回答了两个问题。因为‘边’不是定向的，所以第一个问题和第二个问题互相引用。图 1 示出了带有不定向‘边’的会话窗口，并且没有消息被选择（所有的消息都是纯文本）。因此，如果用户希望突出涉及客户端 2 的消息“是的，但是今天我们要在食堂吃饭吗？”

20 的所有消息，则问题“喂，你们都准备好吃午饭了吗？”以及回答“是的”将被选择。

图 7 示出了带有不定向‘边’的会话窗口，其中客户端 1 和客户端 3 的消息被选择（以斜体字突出）。客户端 1 的消息被间接地从客户端 2 的消息引用，并且客户端 3 被直接地引用。

25 在‘边’具有方向（如图 8 所示）并且同一客户端希望突出所有直接引用客户端 2 的消息的情况下，则只有客户端 3 的回答“是的”被选择（以斜体突出）。

因此，在消息已经被发出之后，该模型允许定向‘边’连接即时消息传送会话中的两条消息。此外，这个模型可以被用来实现与被链接的消息有关的

30 的额外功能，例如对所有引用始发消息的消息执行功能。另外，所提出的模型允许用户通过在生成新消息时指定新消息所回复的预先存在的消息，来同

时生成‘边’和消息。

附图说明

为了促进对本发明的这个和其他方面的更完整的理解,现在将参考附图,

5 仅通过例子的方式来描述本发明的实施例,其中:

图 1 是在即时消息传送系统中简单会话的开始示例;

图 2 是用图形表示并由不定向边链接的会话;

图 3 是通过用箭头表示的定向边链接的两个消息的例子;

图 4 示出了图形上的不定向边;

10 图 5 示出了图形上的两条定向边,该图形上叠加了默示的间接引用;

图 6 示出了具有不定向边的会话窗口,其中没有相关的消息被选择(所有的消息是纯文本);

图 7 示出了具有不定向边的会话窗口,其中,所有从客户端 2 的消息引用的消息被选择(以斜体字突出);

15 图 8 示出了具有定向边的会话窗,其中,只有被从客户端 2 的消息直接引用的消息被选择(以斜体字突出);

图 9 示出了实施例的对象的高层次概括;

图 10 示出了图 9 的交谈对象和相关的对象;

图 11 示出了图 10 的带有‘边’的对象;

20 图 12 示出了生成‘边’的步骤;

图 13 示出了在交谈中发送附加了‘边’的‘交谈消息’;

图 14 示出了删除‘边’的步骤;

图 15 示出了当前的会话窗口;

图 16 示出了在图 15 的窗口内选择第一消息的光标;

25 图 17 示出了移动到第二消息上的图 16 的光标;

图 18 从系统观点示出了生成‘边’的步骤;

图 19 示出了带有‘边’的交谈 14 的会话窗口;

图 20 示出了图 19 的带有另一条‘边’的会话窗口;

图 21 示出了客户端屏幕上的交谈的状态;

30 图 22 示出了相关信息被选择的图 21 的交谈;

图 23 示出了带有‘边’的图 22 的被选择的信息;

图 24 示出了用于删除‘边’的步骤和事件流;

图 25 示出了用于表示由边链接的消息的字体格式编排;

图 26 示出了用于表示由边链接的消息的数字; 和

图 27 示出了用于表示边的弯曲箭头。

5

具体实施方式

针对图 9 描述本发明的优选实施例。优选实施例包含: 服务器 10; 客户端 12; 和各交谈对象 (交谈 14A 到交谈 14N)。服务器 10 存在于服务器计算机 16 上, 并通过网络与在各远程用户计算机 18A 到 18M 上的客户端 12 (图 9 中仅示出了一个) 进行通信。所有的通信经过服务器 10, 并且通过服务器 10 生成和管理交谈。服务器 10 维护交谈的集合, 并保持每个交谈 14 的确定版本。在用户计算机 18A 上, 客户端 12 向用户展示图形用户界面 (GUI), 即时消息传送对话可以通过该图形用户界面进行。和服务器 10 类似, 每个客户端 12 维护与用户所参加的即时消息传送对话集对应的一个或更多个交谈的集合。客户端 12 所维护的每一个交谈 14A 到交谈 14N 是服务器 10 所维护的交谈 14A 到 14N 的拷贝。交谈 14 被更新, 以使它与服务器 10 所保持的对应的拷贝一致; 当对服务器 10 所维护的交谈对象做出修改时, 服务器 10 将通知各个客户端 12。

客户端-服务器实施例仅仅是一种可能性。在对等的实施例中, 将存在竞争的问题, 所以如果两个客户端 12 试图在大致相同的时间发送消息, 则总体上系统必须做出有关于首先发送哪一个的决定 (因为消息是以时间顺序显示给用户的)。由于该决定由服务器做出, 优选客户端-服务器实施例。

图 10 示出了带有被关联的用户对象 (用户 20A 到用户 20O) 和被关联的‘交谈消息’对象 (‘交谈消息’ 22A 到 ‘交谈消息’ 22P) 的交谈 14。交谈 14 包含 ‘唯一交谈 ID’ 19 属性。‘唯一交谈 ID’ 19 在整个系统中唯一地标识交谈 14, 并被其他对象使用来引用交谈 14。用户 20A 到用户 20O 中的每一个均代表当前正在参加交谈的实际用户。用户 20A 到用户 20O 中的每一个均具有 ‘唯一用户 ID’ (24A 到 24O) 属性和 ‘名称’ (26A 到 26O) 属性。

‘唯一用户 ID’ 24A 是整个系统中用于标识用户的唯一 ID, 并被其他的对象使用来引用该用户。‘名称’ 26A 属性是交谈中显示给其他用户的用户名称。

‘交谈消息’ 22A 代表交谈中的单个消息, 该消息可以是对话的单个行,

每一行被回车分隔开。交谈 14A 的确定拷贝由服务器 10 保存，只有响应客户端所做出的请求，才操作确定拷贝。然后，对此确定拷贝所做出的任何改变都被复制到客户端 12 上的交谈 14A 的本地拷贝上。每一个交谈 14A 将具有‘交谈消息’ 22A 到 22P 的序列。当用户把文本键入 GUI 上的文本输入框并将其发送时，交谈 14 生成新的‘交谈消息’ 22A。‘交谈消息’ 22A 包含下列属性：‘唯一消息 ID’ 30A；‘文本’ 32A；‘时间标记’ 34A；和‘用户 ID’ 36A。‘交谈消息’ 22B 到 22P 具有对应的属性。

‘唯一消息 ID’ 30A 是整个系统中用于标识‘交谈消息’ 22A 的唯一 id。‘唯一消息 ID’ 30A 被其他的对象使用来引用‘交谈消息’ 22A。‘交谈消息’ 22A 的‘文本’ 32A 是用户键入 GUI 上的文本输入框内的文本。‘时间标记’ 34A 是服务器 10 上生成‘交谈消息’ 22A 的时间。服务器 10 上的‘交谈消息’ 22A 到 22P 被以‘时间标记’ 34A 到 34P 的升序储存。‘用户 ID’ 36A 是负责生成‘交谈消息’ 22A 的用户 20A 的唯一 ID。

图 11 是图 10 的扩展。它扩展了‘交谈消息’ 22A 到 22P，以便包括‘边’ 38A 到 38Q 的集合。‘边’由两个或更多个‘交谈消息’ 22A 到 22P 以及用于结合它们的图形链接的类型来定义。用户可以生成‘边’ 38A 来指示两个‘交谈消息’ 22A、22B 之间的关系，从两个‘交谈消息’的如 GUI 上所显示的初始位置来看，这样的关系可能不清晰。例如，如果两个是的/不是问题被不同的用户问起，并且第三个用户随后回答了“是的”，则第三个用户可以生成‘边’来指示他们正在回答哪一个问题。在优选实施例中，‘边’链接两条‘交谈消息’。但是，‘边’可以被扩展，以使它引用另一条‘边’或者引用多于两条的‘交谈消息’。

‘边’ 38A 包含下列属性：‘唯一边 ID’ 40A；‘第一消息 ID’ 42A；‘第二消息 ID’ 44A；和‘边类型’ 46A。‘边’ 38B 到 38Q 具有对应的属性。

如果引用另一条‘边’ 38Q，则‘边’ 38A 还包含边名称属性（未示出）。如果‘边’能够引用多于两条‘交谈消息’，则‘边’包含消息 id 的集合（例如数组或向量）。

‘唯一边 ID’ 40A 是整个系统中用于标识‘边’ 38A 的唯一 id；它被其他对象使用来引用‘边’ 38A。‘第一消息 ID’ 42A 指‘交谈消息’ 22A，这条‘边’ 38A 是从‘交谈消息’ 22A 关联的。‘第二消息 ID’ 44A 指这条‘边’指向的‘交谈消息’ 22P。‘边类型’ 46A 属性定义了‘边’ 38A 是有方向性

的还是没有方向性的。

像‘交谈消息’和‘用户’一样，‘边’与特定的交谈相关联。这样，对每个交谈，存在着交谈 14 中的每一条‘边’的拷贝和该交谈中的每一条‘交谈消息’的拷贝。任何‘边’对象或‘交谈消息’对象的确定拷贝存在于服务器 10。如果客户端 12A 具有交谈 14A 的被关联的拷贝，则客户端 12A 也必须保持每条‘边’的拷贝和每个‘交谈消息’的拷贝。

优选实施例实施了实时交谈系统的已知操作：连接到服务器 10 并登录；与服务器 10 断开连接；生成交谈 14A 到 14N；邀请其他的用户加入交谈；接受加入交谈的邀请；当在交谈中时发送消息；和离开交谈。优选实施例定义了额外的操作：在交谈 14A 中生成‘边’ 38A；把附加了边的消息发送到交谈 14A；和删除‘边’。

图 12 描述了在交谈中生成‘边’ 38A 的步骤：

步骤 1201。用户 1 使用客户端 12A GUI 来指定要被‘边’ 38A 连接在一起的两条‘交谈消息’ 22A、22B。这通过 GUI 机制完成，在这部分不讨论此 GUI 机制；

步骤 1202。发起客户端 12A 发送‘边请求’消息到服务器 10。该消息指定交谈 14A 的 ID 和要被连接在一起的两条‘交谈消息’ 22A 和 22B 的消息 ID；

步骤 1203。服务器 10 生成包含两条被指定的‘交谈消息’的‘唯一消息 id’的‘边’ 38A；

步骤 1204。服务器 10 把‘边’ 38A 关联到被指定的交谈 14A；

步骤 1205。服务器 10 向交谈 14A 中所有的客户端 12A 到 12N（包括发送边请求的那个客户端）广播‘边通知’。‘边通知’包含‘边’ 38A 的属性；

步骤 1206。接收到这个‘边通知’的每一个客户端 12 生成‘边’ 38A 的本地拷贝；和

步骤 1207。每一个客户端 12 随后把此新生成的边 38A 的本地拷贝添加到其自己的交谈 14A 的本地拷贝中。

图 13 描述了客户端 12A 生成新‘交谈消息’ 22B 以及‘边’ 38A 的步骤，‘边’ 38A 把该（新）‘交谈消息’ 22B 链接到另一个（预先存在的）‘交谈消息’ 22A。

步骤 1301。用户 1 指定 a) 新消息的文本和 b) 新消息要被连接到的预

先存在的消息。这通过 GUI 机制完成，不讨论 GUI 机制；

步骤 1302。客户端 12A 发送消息‘带有边请求的交谈消息’消息到服务器 10。该消息指定交谈 14A 的‘唯一交谈 ID’ 19、新‘交谈消息’ 22B 的文本和新‘交谈消息’ 22B 要被‘边’ 38A 连接到的预先存在的‘交谈消息’

5 22A 的 ID；

步骤 1303。服务器 10 生成带有指定文本的新‘交谈消息’ 22B；

步骤 1304。新生成的‘交谈消息’ 22B 与服务器 10 的交谈 14A 的拷贝相关联；

步骤 1305。服务器 10 还生成连接被指定的预先存在的‘交谈消息’ 22A
10 和新生成的‘交谈消息’ 22B 的新‘边’ 38A；

步骤 1306。新生成的‘边’ 38A 被添加到服务器 10 的交谈 14A 的拷贝中；

步骤 1307。服务器 10 还发送‘交谈消息通知’消息到交谈 14A 中的每一个客户端 12A 到客户端 12N，注意，包括客户端 12A。这个消息指定了用于完整地描述‘交谈消息’的充足信息。
15

步骤 1308。接收到‘交谈消息通知’的每一个客户端 12A 到客户端 12N 生成新‘交谈消息’ 22B 的本地拷贝。

步骤 1309。每一个客户端 12 把新生成的‘交谈消息’ 22B 的本地拷贝添加到其自己的该交谈的本地拷贝中。

步骤 1310。服务器 10 向交谈 14A 中的客户端 12 广播‘边通知’消息(包括发送‘带有边请求的交谈消息’的那个客户端)。
20 ‘边通知’消息包含用于描述边 38A 的充足的信息。

步骤 1311。接收到这条‘边通知’消息的每一个客户端 12A 到客户端 12N 生成‘边’ 38A 的本地拷贝。

步骤 1312。然后，每一个客户端 12 把新生成的‘边’ 38A 的本地拷贝与其自己的交谈 14A 的本地拷贝相关联。

图 14 描述了实施例删除‘边’的步骤。

步骤 1401。用户 1 使用客户端 12 GUI 来指定交谈 14A 中要被删除的现有‘边’ 38A。这通过 GUI 机制完成，不讨论 GUI 机制；

步骤 1402。客户端 12A 发送‘删除边请求’消息到服务器 10。这个消息指定了交谈 14A 的 ID 和要被删除的‘边’ 38A 的 ID；
30

步骤 1403。服务器 10 命令交谈 14A 删除被关联的‘边’ 38A;

步骤 1404。交谈 14A 删除‘边’ 38A;

5 步骤 1405。服务器 10 向该交谈中所有的客户端 12 (包括发送‘删除边请求’消息的那个客户端) 广播‘删除边通知’消息。‘删除边通知’消息包含‘边’ 38A 的 ID;

步骤 1406。接收到该‘删除边通知’消息的每一个客户端 12A 到客户端 12N 命令其被关联的本地交谈 14A 去除各自的‘边’ 38A; 和

步骤 1407。然后每一个交谈 14A 删除其各自的‘边’ 38A。

10 在下列例子中, 边被图形化地表示为相关消息之间的箭头。稍后描述对于此的其他可能性, 例如使用不同颜色的文本来代表相关的消息。

图 15 的交谈 14A 窗口被显示在参加当前交谈的所有客户端 (与客户端 12A、客户端 12B 和客户端 12C 对应的客户端 1、客户端 2 和客户端 3) 的屏幕上。交谈中有由客户端 1 和客户端 2 提出的两个问题, 这导致来自客户端 3 的回答是不确定的。不清楚他回答了是在下午 1 点有空还是他在下午 2 点有空开会。当来自一个用户 (客户端 2) 的问题被输入到交谈 14A, 而另一个用户 (客户端 3) 正处在键入对较早问题的回答之中时, 经常出现这种场景。这种不确定性可以通过由客户端 3 生成位于问题之一和他们的回答之间的‘边’ 38A, 从而指出他们正在回答哪个问题来解决。这可以通过由客户端 3 选择他们已经回答的问题并把问题“拖”到回答上来完成, 如图 16 和图 17 所示。然后, 系统知道在该问题和回答之间生成‘边’ 38A。

图 16 示出在鼠标指针的相应运动之后, 光标位于‘边’ 38A 要引用的第一消息上面。用户按下鼠标左键并保持不放, 以使第一消息随着鼠标指针移动, 并且第一消息被移动到第二消息上面。图 17 示出从第一消息移动到边 38A 所引用的第二消息上面之后的光标。然后释放鼠标左键。

25 图 18 示出了在整个系统中如何进行边生成。

现在, 可以在交谈客户端上相应地表示边 (在这些例子中, 边被图形化地表示为消息之间的箭头)。

步骤 1801。发送‘边请求’以在消息和回答之间生成‘边’ 38A。

步骤 1802。在边的集合中生成‘边’ 38A。

30 步骤 1803。‘边通知’被发送到交谈中所有的客户端。

图 19 示出交谈的新状态, 该‘边’指示客户端 3 正在回答哪一个问题。

然后客户端 3 可以像平常一样继续会话，或者也可以使用‘边’手段来回答其他问题，如图 20 所示，其中生成另一条边来指示对客户端 2 的问题的回答。

现在参考图 21 到图 23 描述多于两个消息的链接，这些消息属于会话内的某个话题。如果交谈 14A 包含来自一个会话内的几个不同话题的分散消息，
5 则能够使用‘边’来突出涉及会话的一个话题的所有消息，这些消息可以与会话的剩余部分分开保存。图 21 示出了具有多个消息的典型的交谈屏幕。客户端 1 想保存这个会话中被选择的消息。可以使用‘边’来链接会话的有用部分。这可以通过选择消息的多个‘有用’部分来完成，一般地可以通过一个人在用鼠标光标选择消息时按下 CTRL 键来完成。一旦所有需要的消息被
10 选择，则用户可以执行把‘边’添加到所有被选择的消息上的操作，见图 22。图 23 中示出此操作的结果。

然后，被选择的消息可以与交谈的其余部分中的剩余消息分开保存。交谈中的任何客户端能够简单地突出初始问题，并挑选“保存相关消息”或类似的操作。问题，以及所有（直接地或间接地）链接到该问题的消息，都将被保存。希望保存交谈的客户端不一定需要在相关消息链中选择第一消息。
15 如果他们只需要从会话中的某个点开始的信息，则他们能简单地突出该链中他们期望选择的消息，并选择‘保存相关消息’。根据边的方向，这将只保存从那个点开始的消息，以及从该消息直接或间接地链接的所有消息（即，不能沿着‘边’向后进行）。

20 通过在屏幕上选择边并按下删除键（或者从客户端 GUI 的菜单上选择‘删除’），可以简单地删除边。图 24 描述了删除‘边’所包含的步骤。

步骤 2401。把带有要被删除的‘边’的 ID 的‘删除边请求’从客户端 12 发送到服务器 10；

步骤 2402。服务器 10 从‘边’的集合中删除该‘边’；和

25 步骤 2403。‘删除边通知’，以及‘边’ ID，被从服务器 10 发送到该交谈中所有客户端 12，并且客户端 12 从它们各自的交谈 14 的拷贝中以及从屏幕上适当地去除‘边’。

除了使用如前面图中所示的箭头在客户端 GUI 上表示边以外，还有几种其他的可能性。方法包括使用不同于其他消息的颜色（未示出）；对用‘边’
30 链接的消息使用不同的字体格式编排（图 25）；使用数字引用相关的消息（图 26）；以及使用弯曲箭头而非直边的箭头（图 27）。

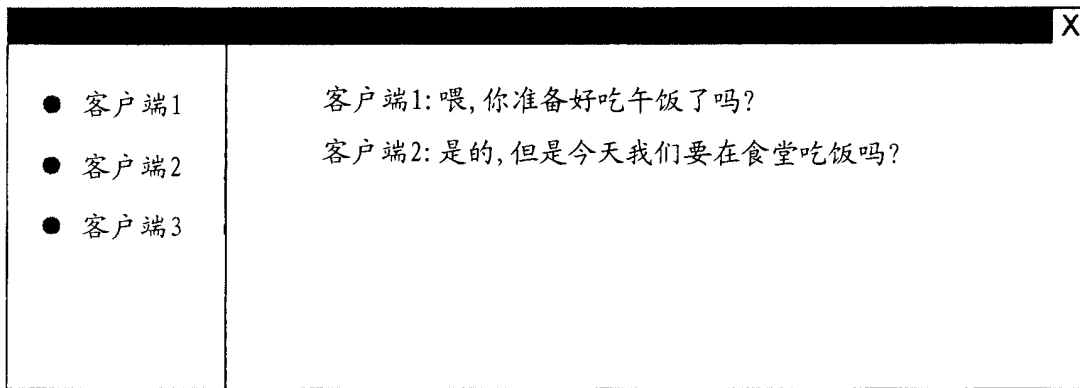


图 1

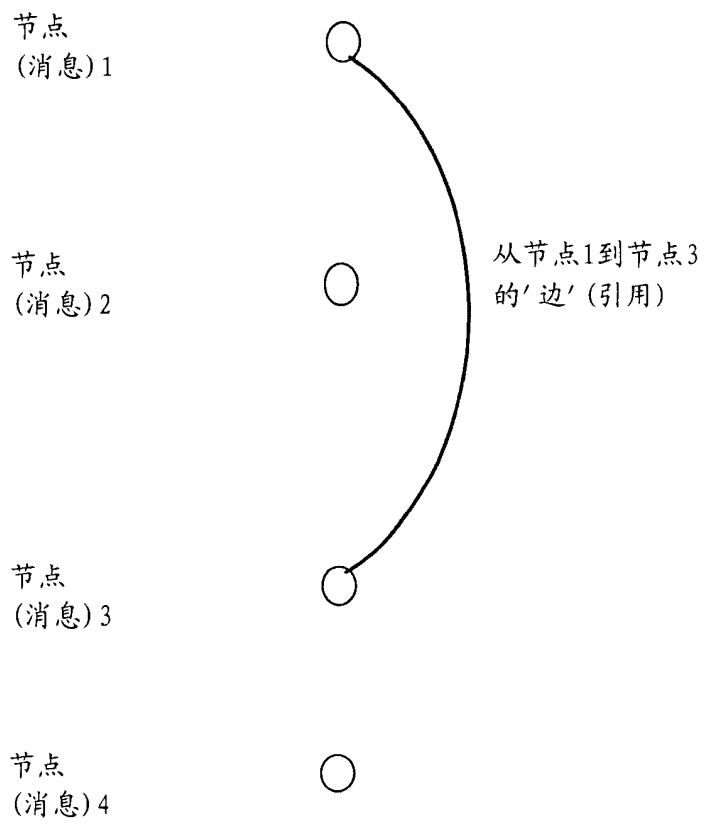


图 2

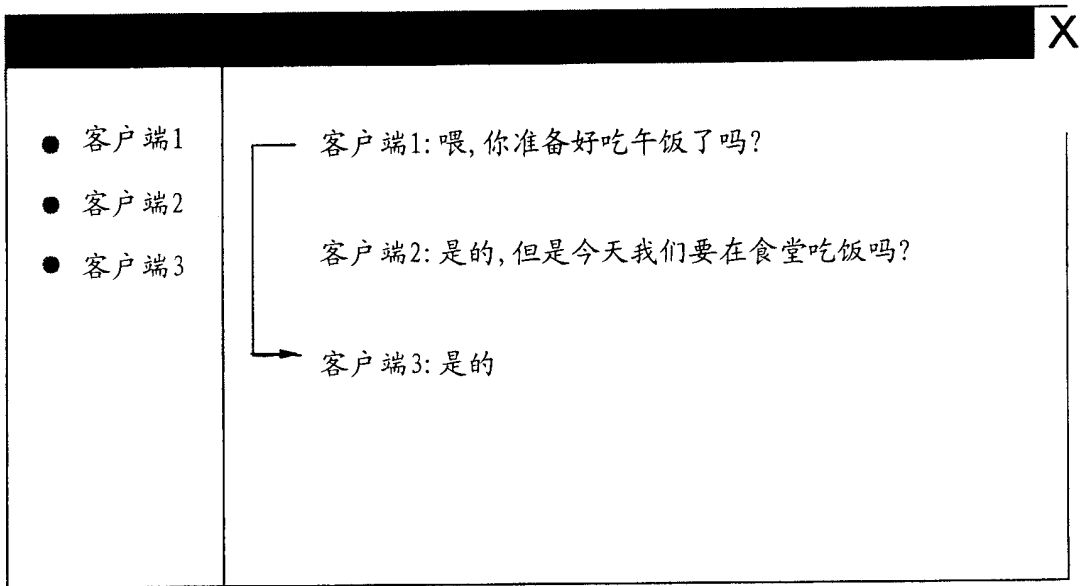


图 3

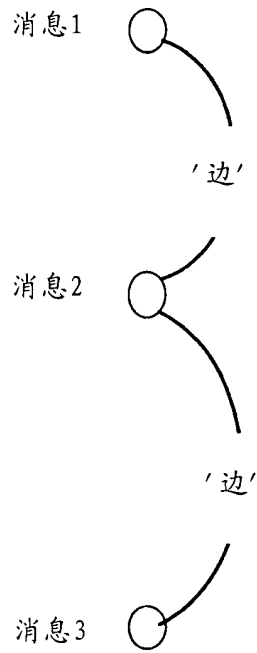


图 4

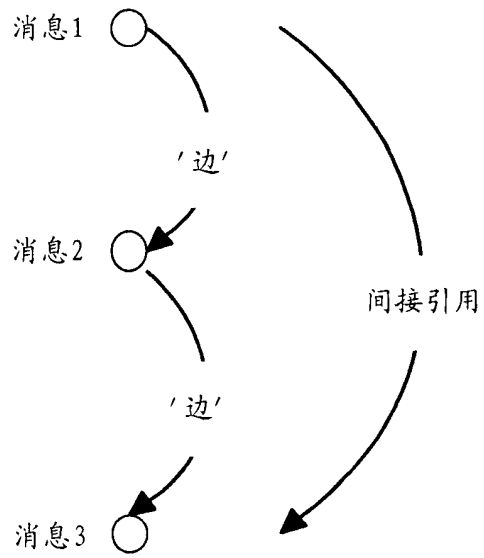


图 5

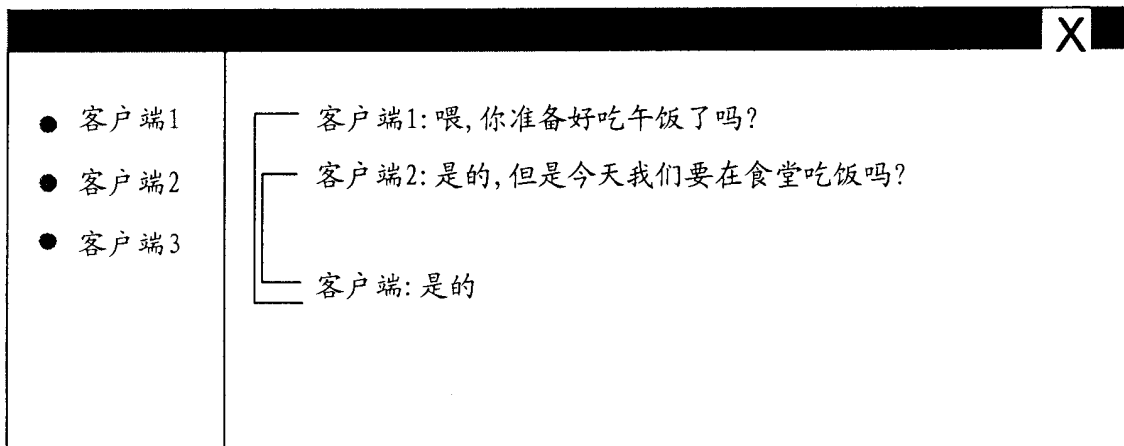


图 6

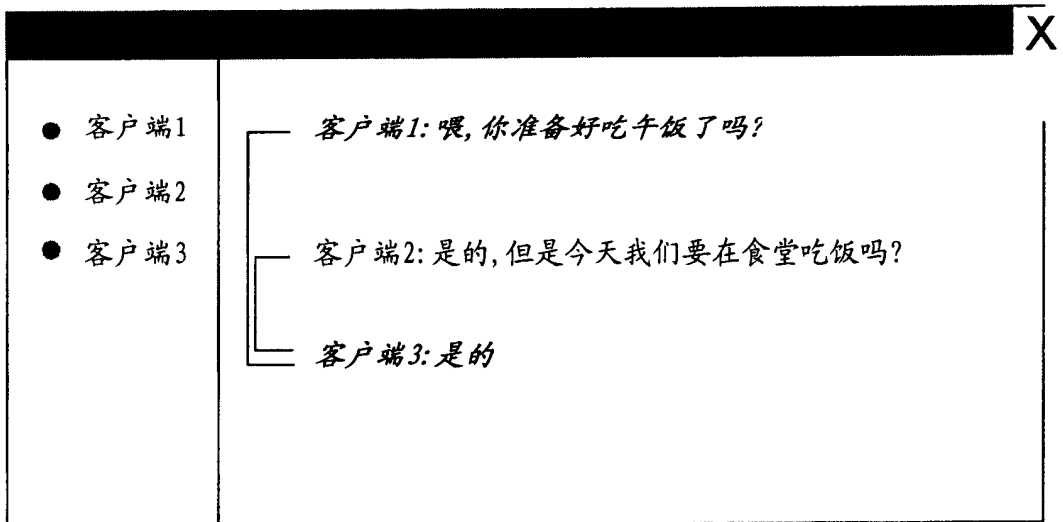


图 7

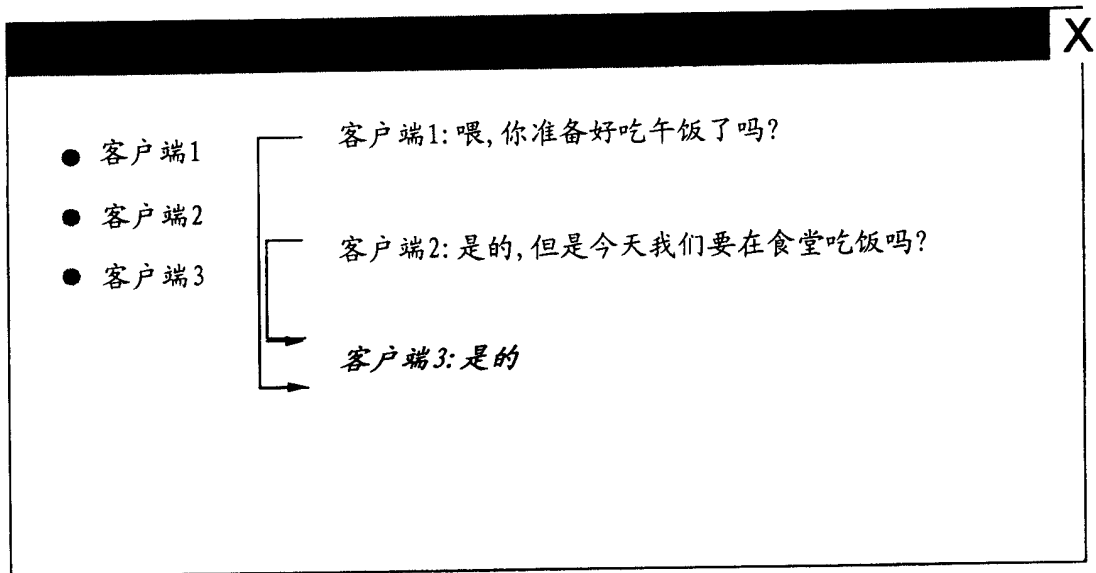


图 8

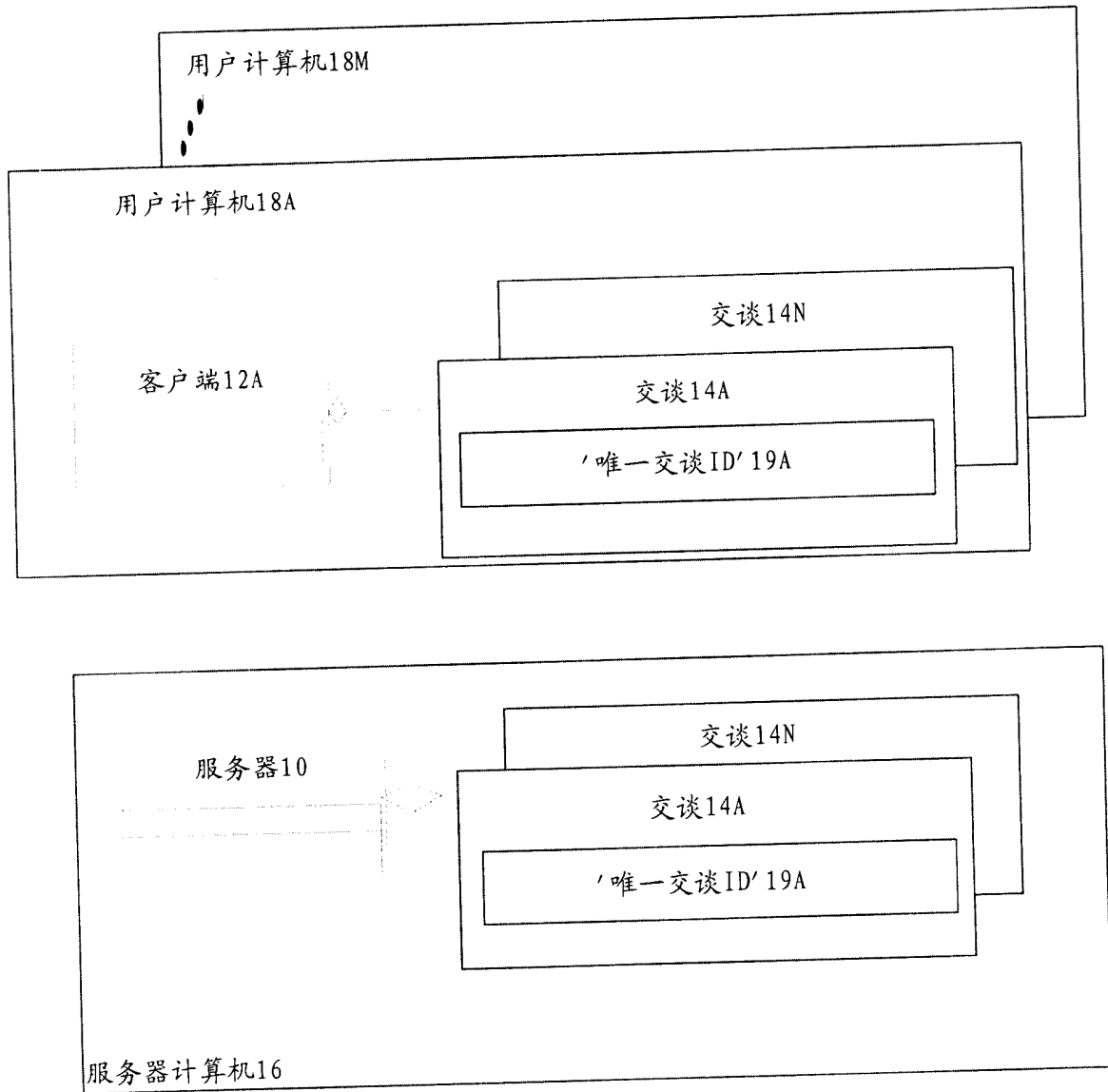


图 9

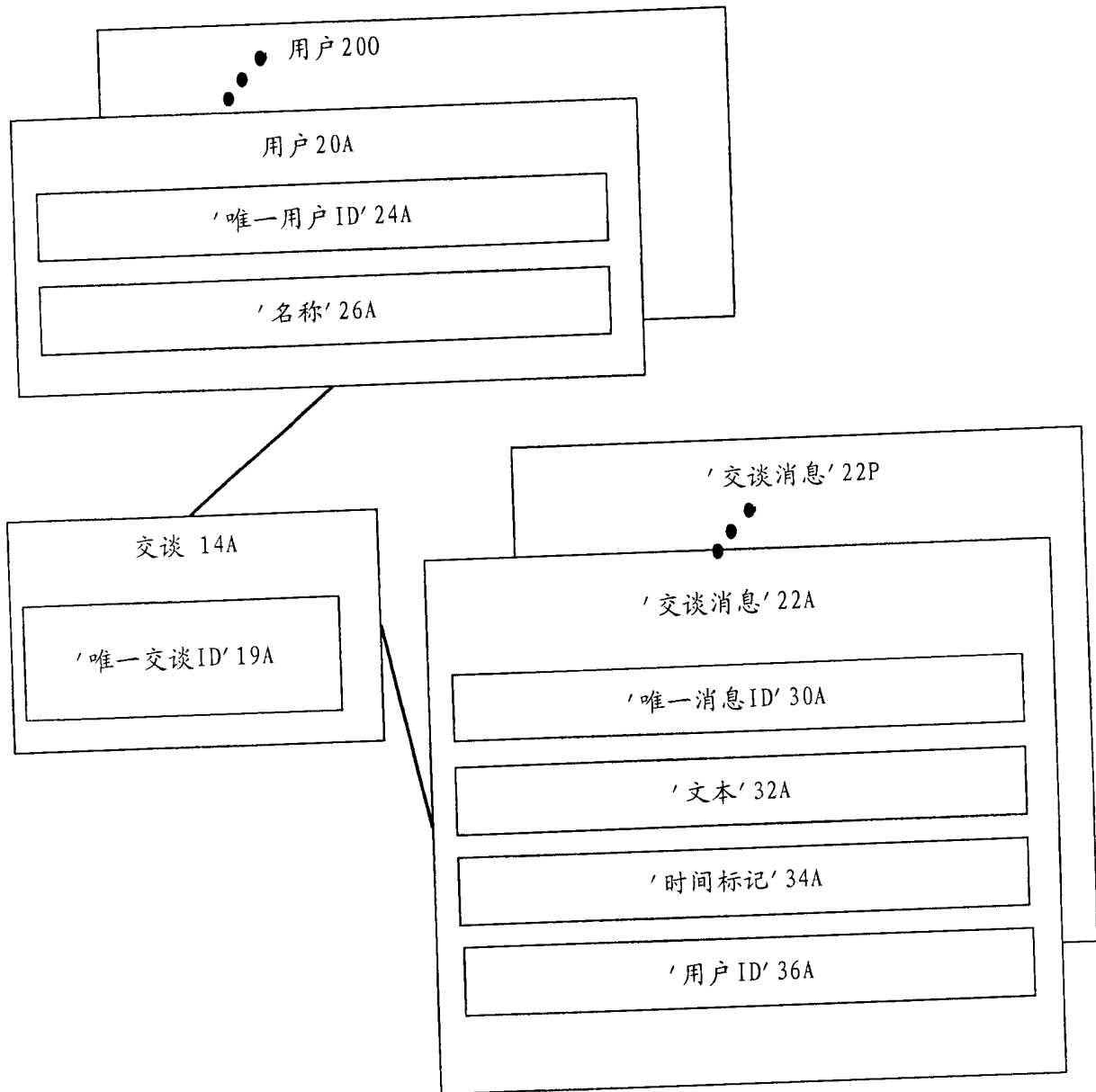


图 10

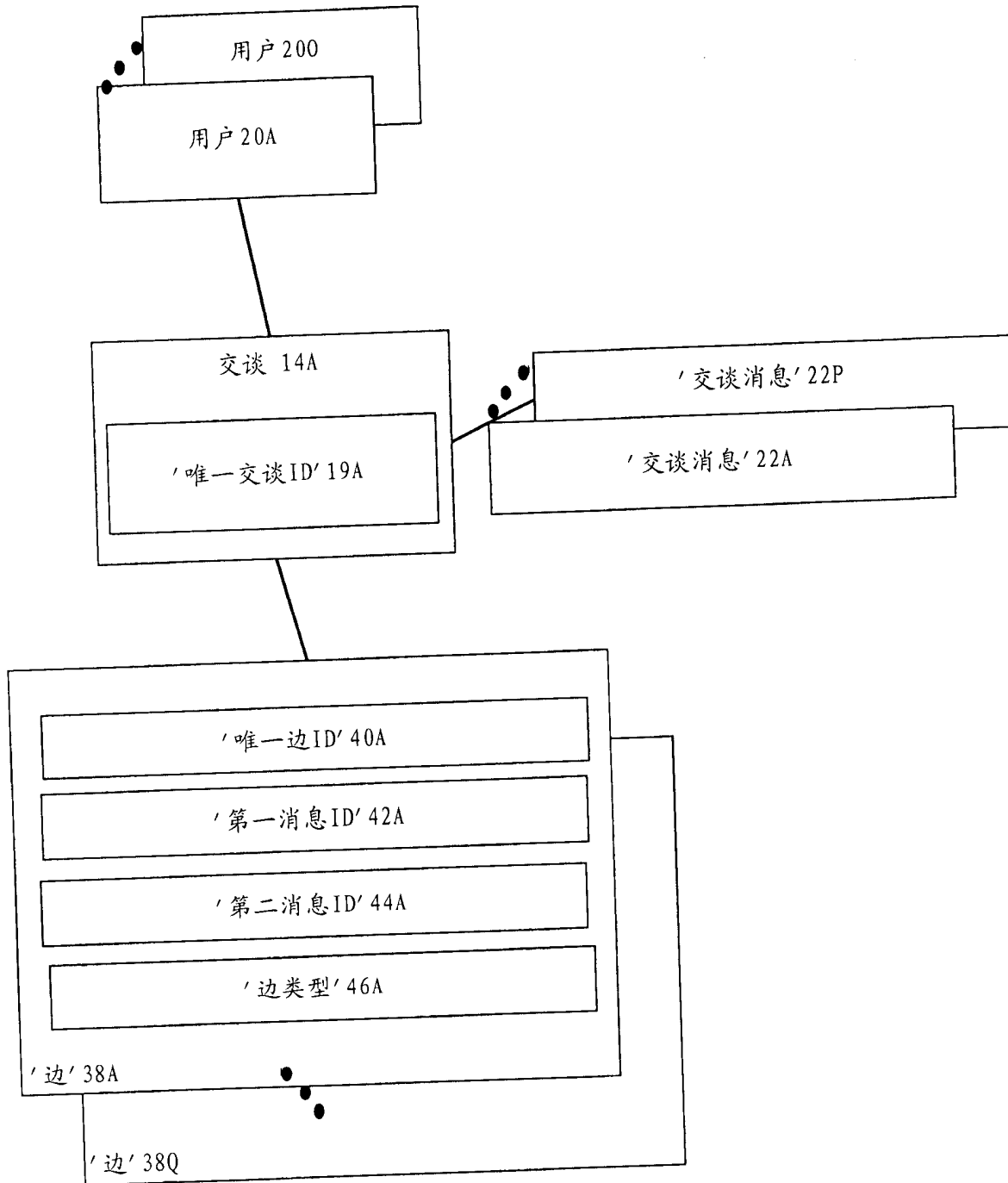


图 11

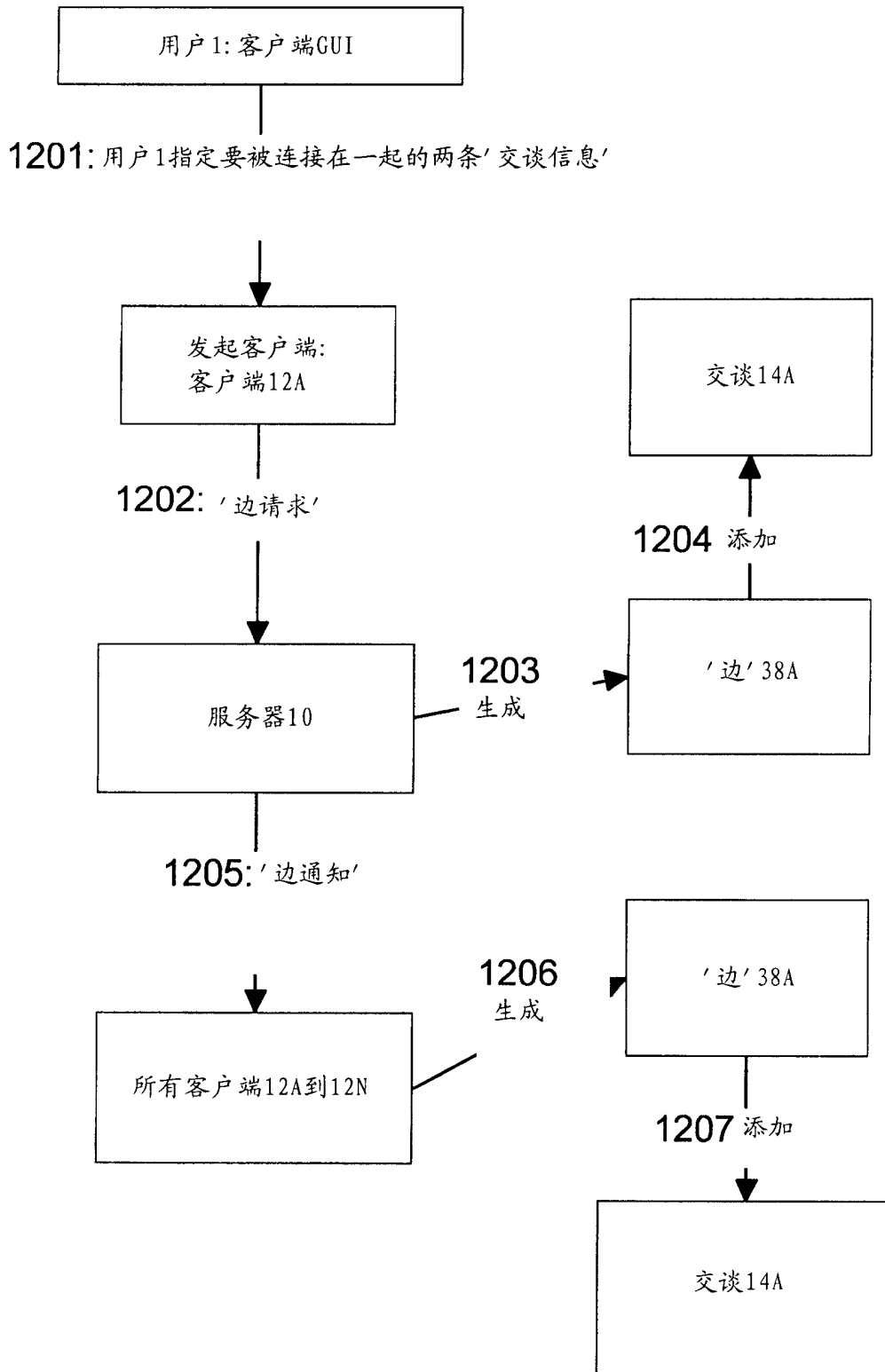


图 12

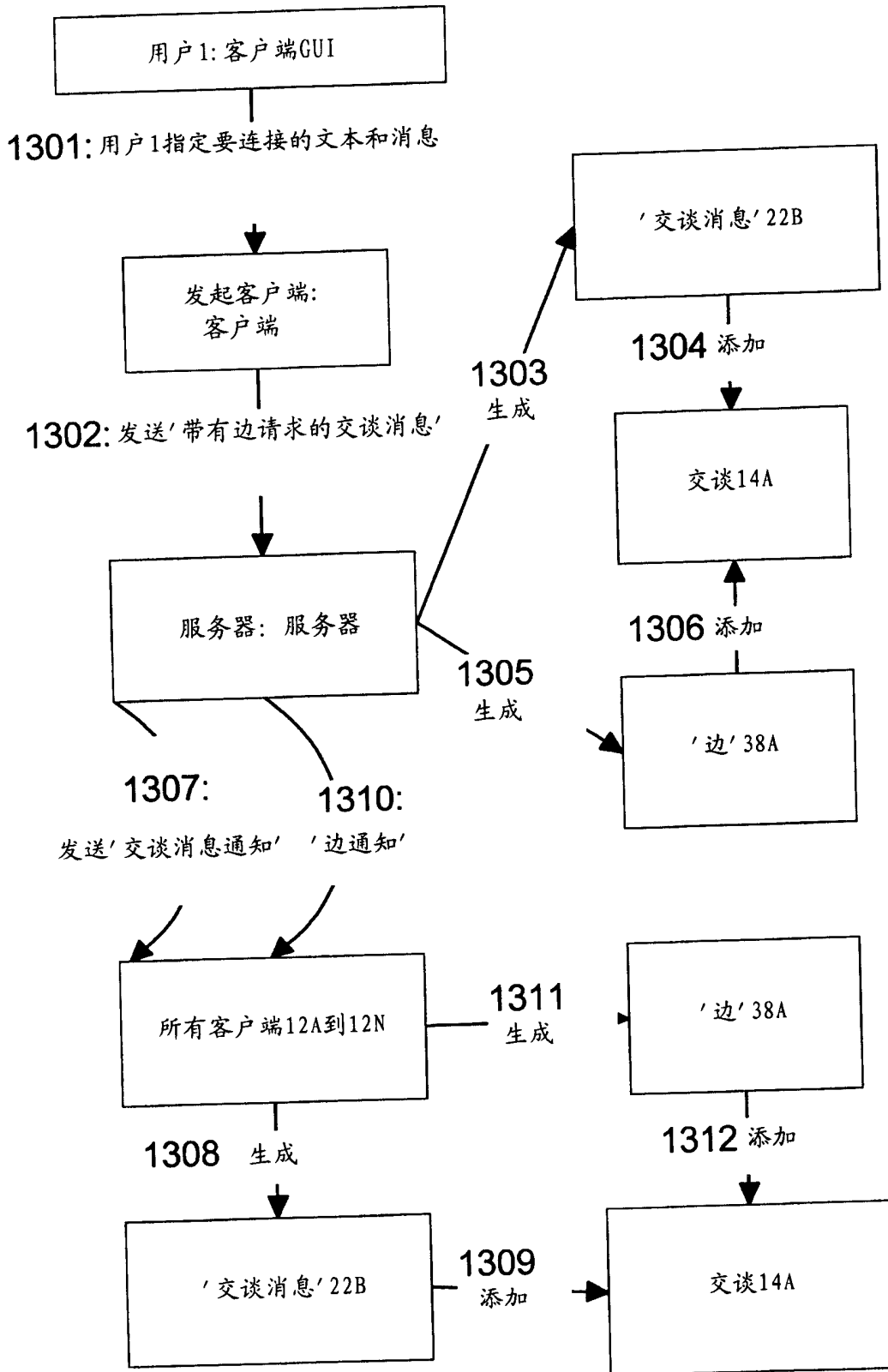


图 13

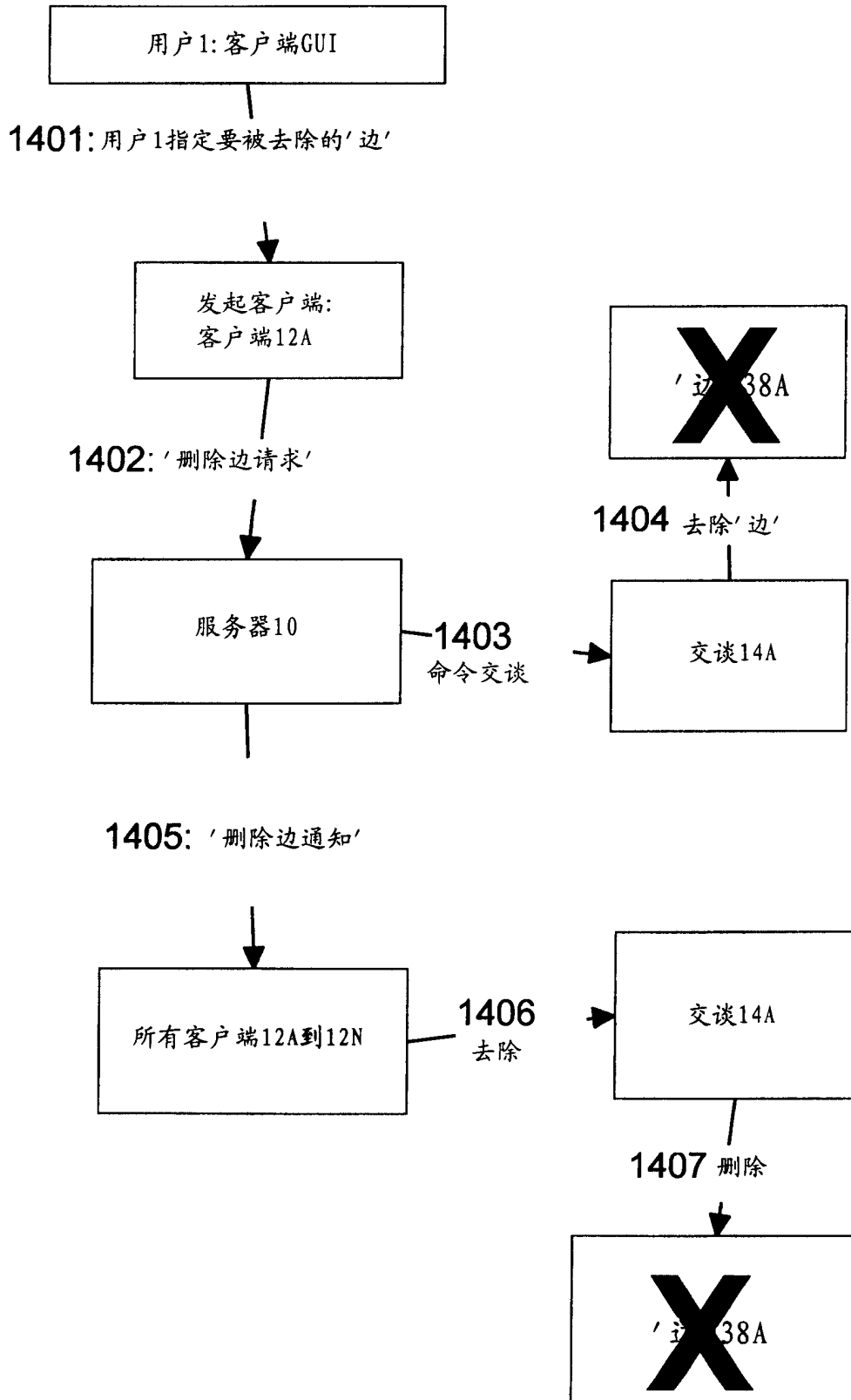


图 14

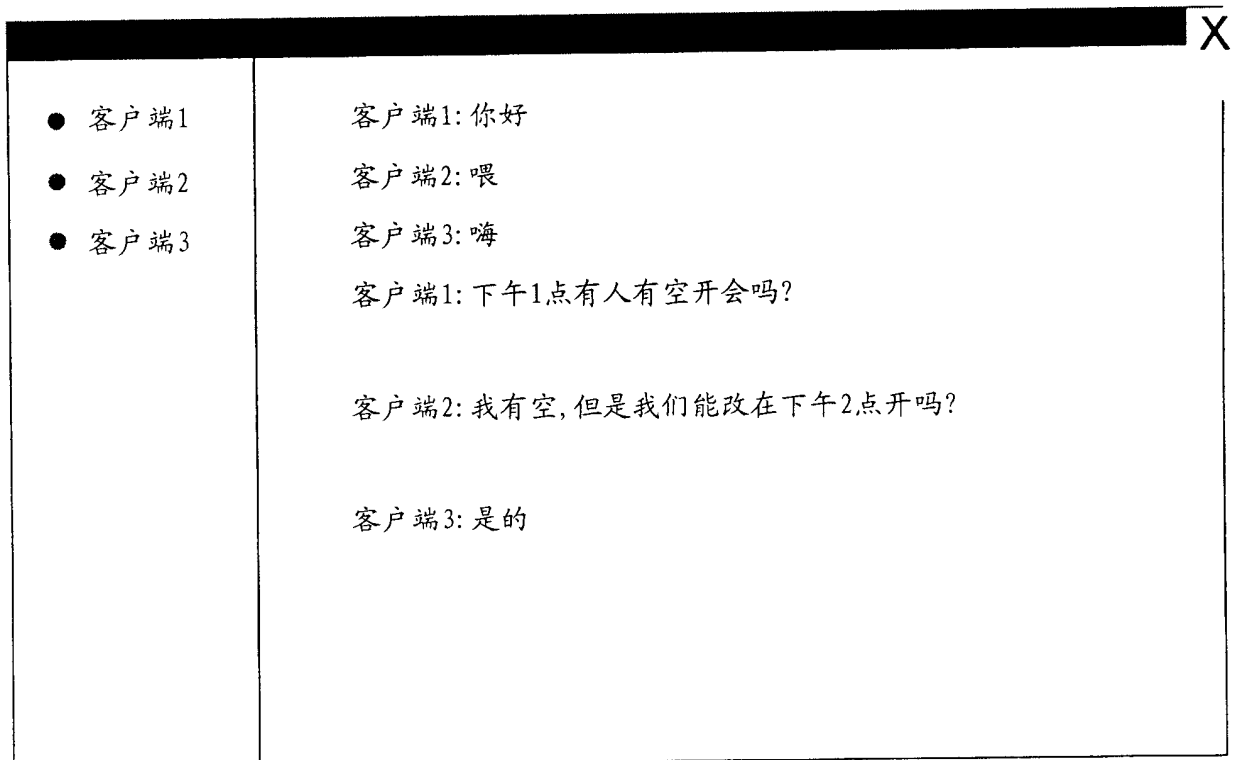


图 15

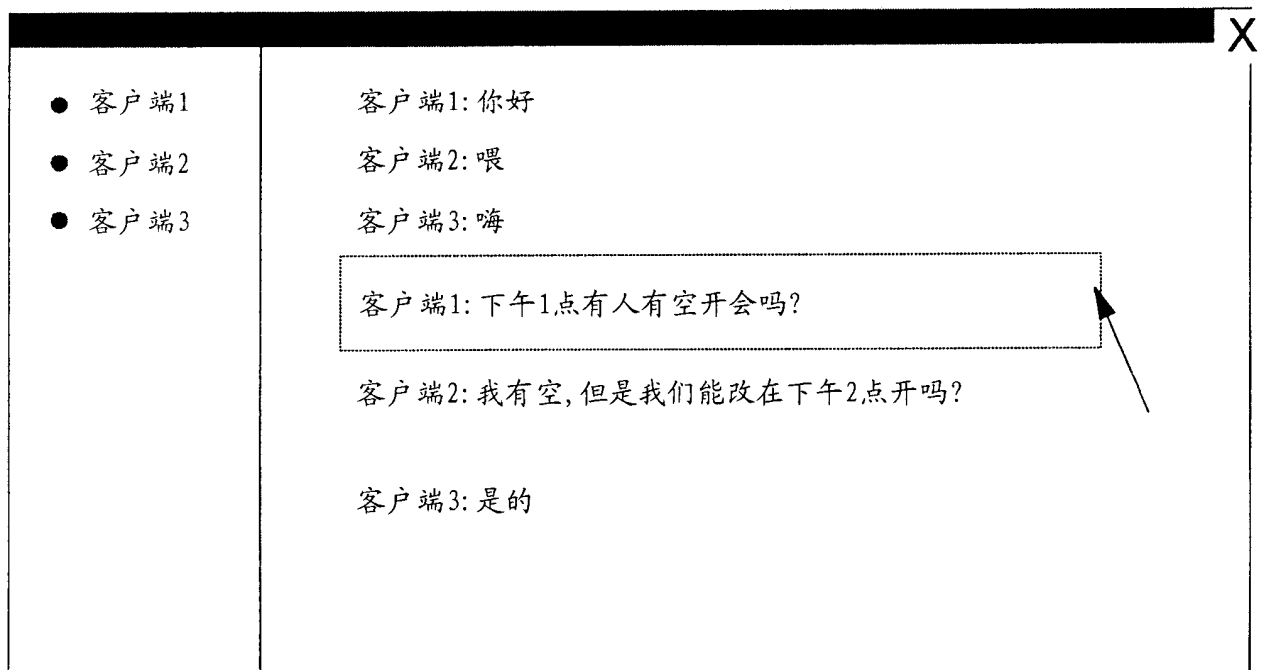


图 16

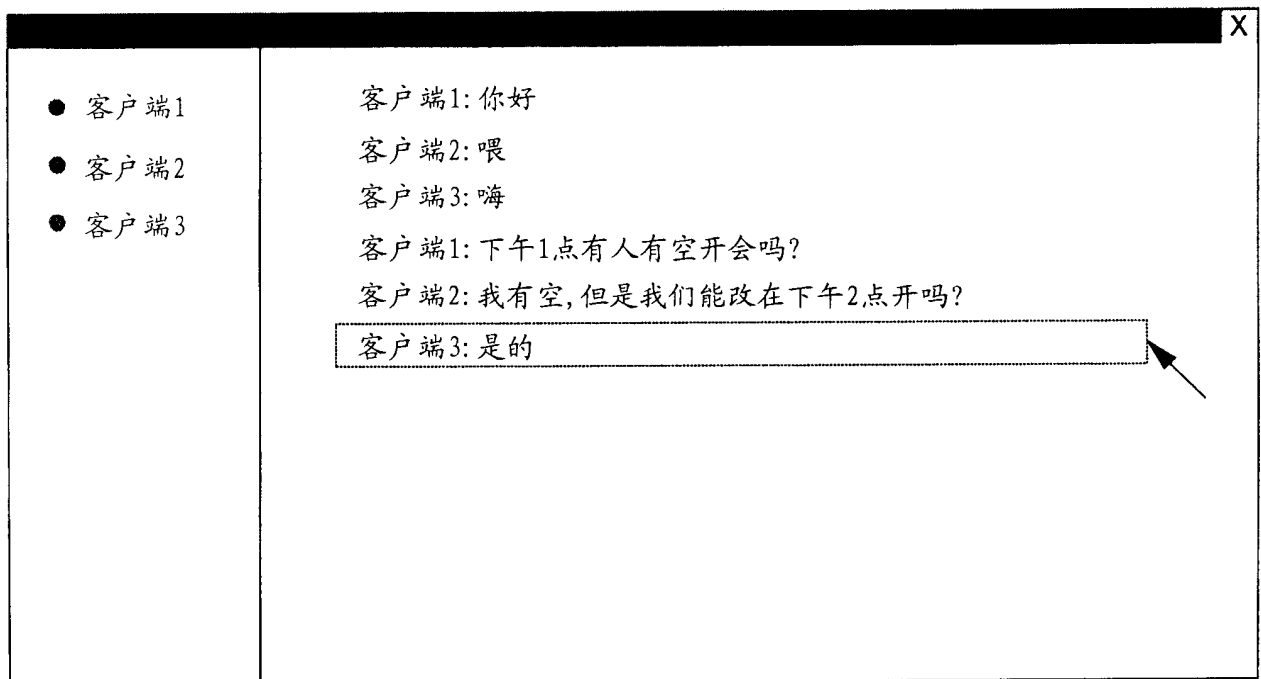


图 17

1803 '边通知'被发送到交谈中的所有客户端。客户端在屏幕上适当地表示'边'。

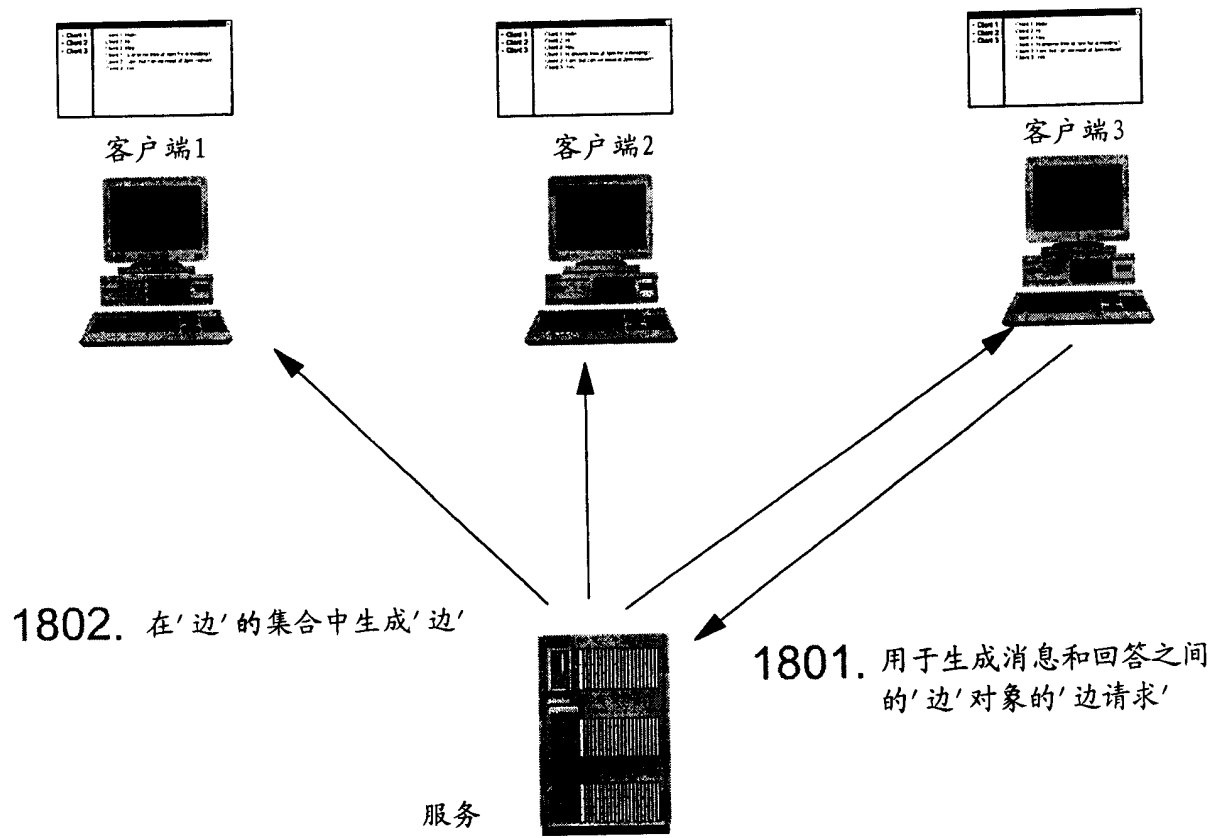


图 18

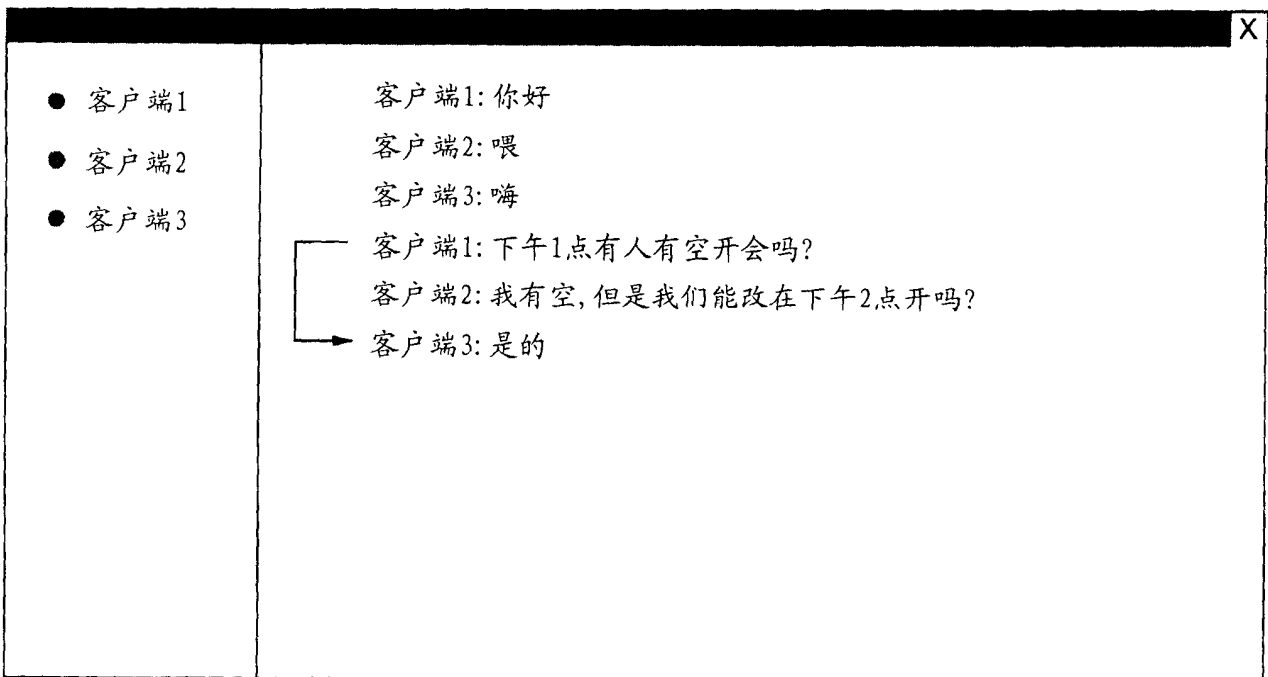


图 19

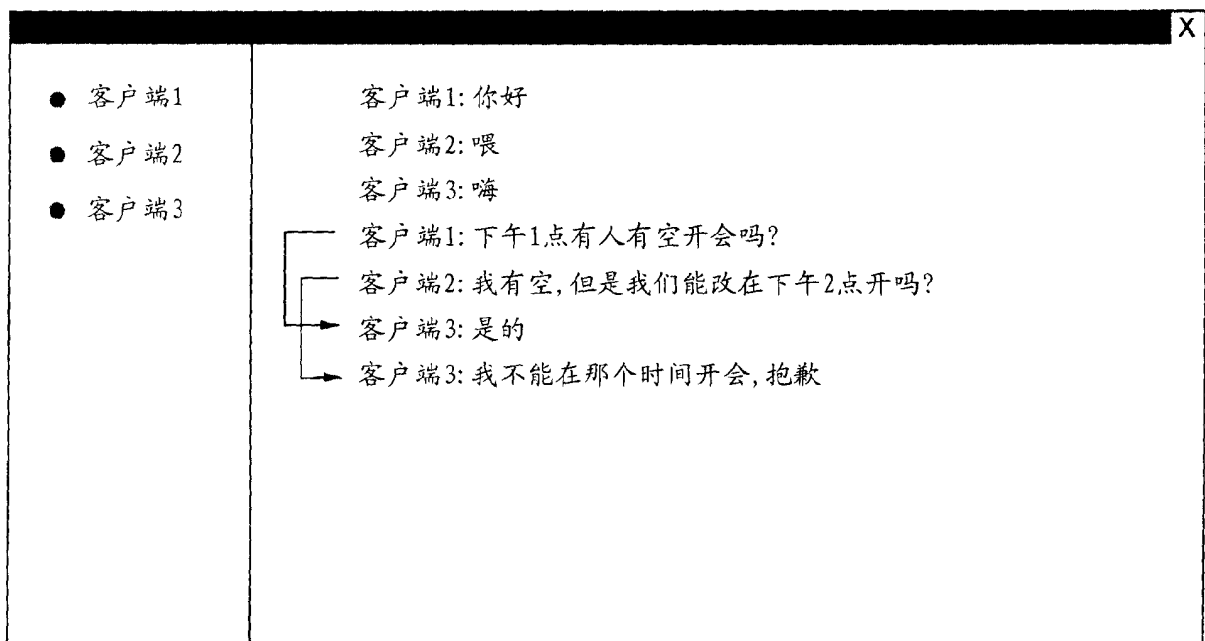


图 20

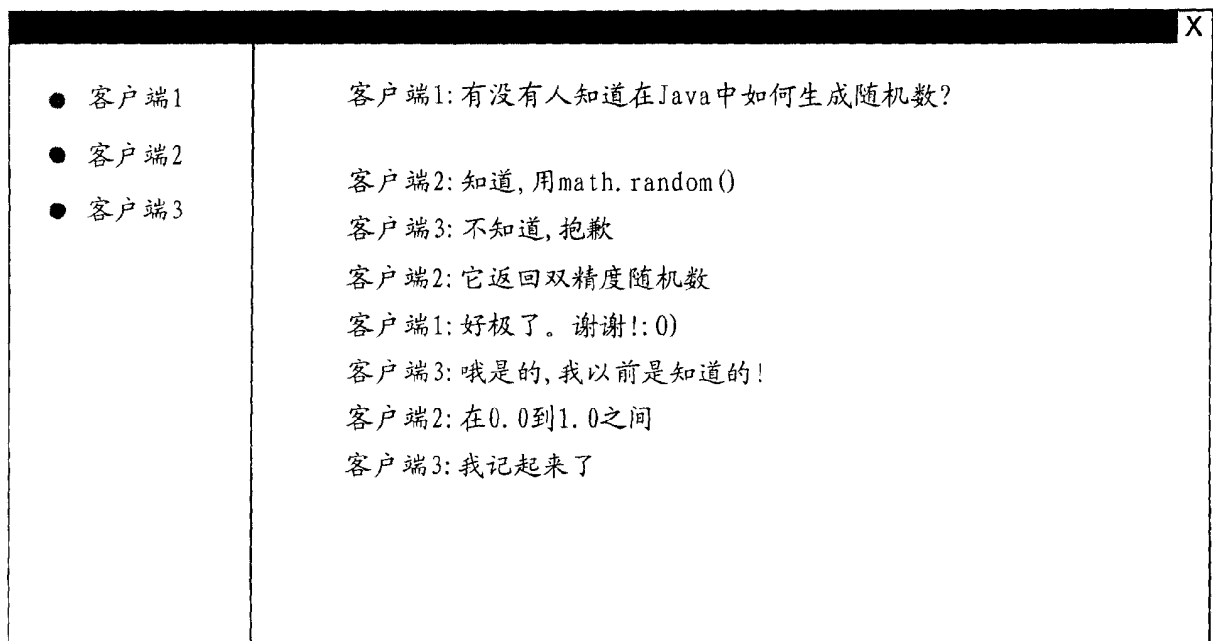


图 21

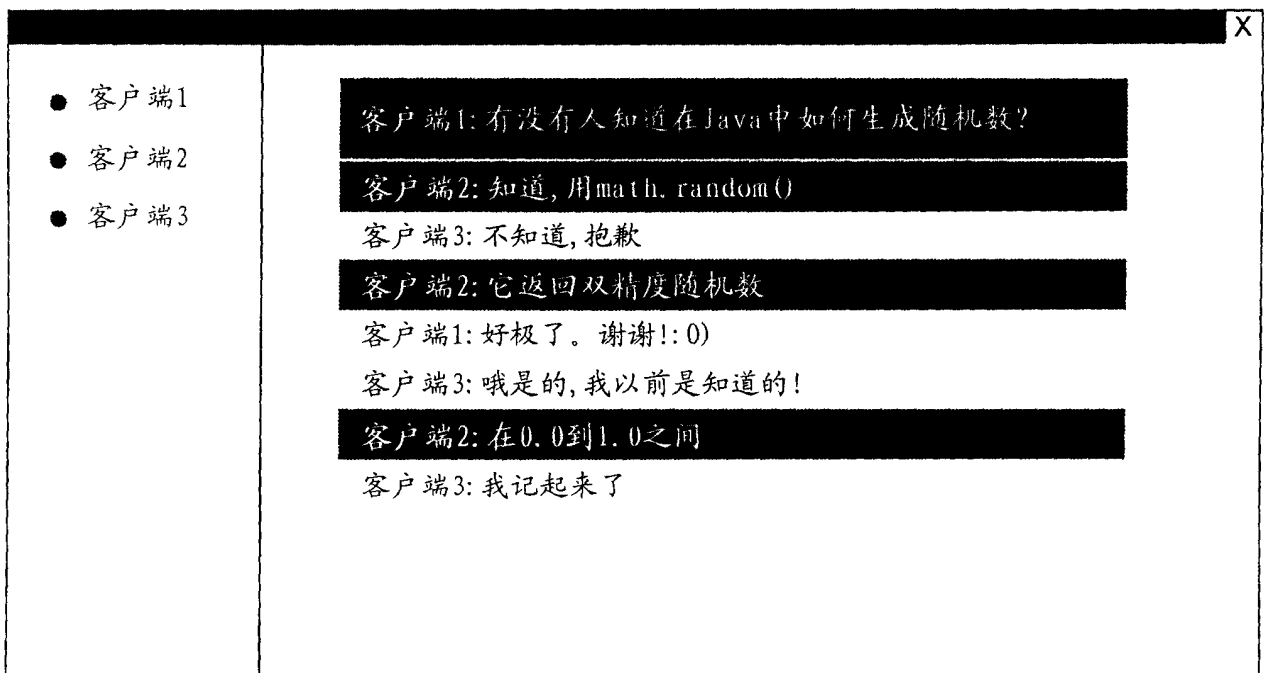


图 22

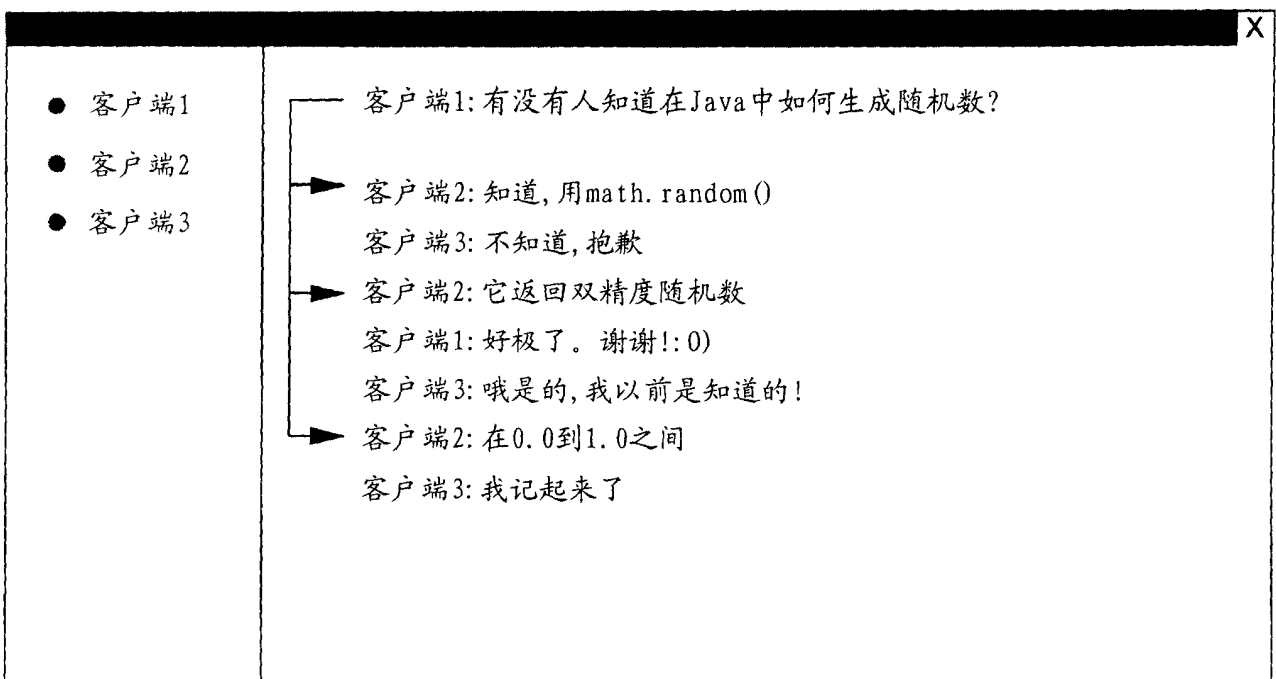
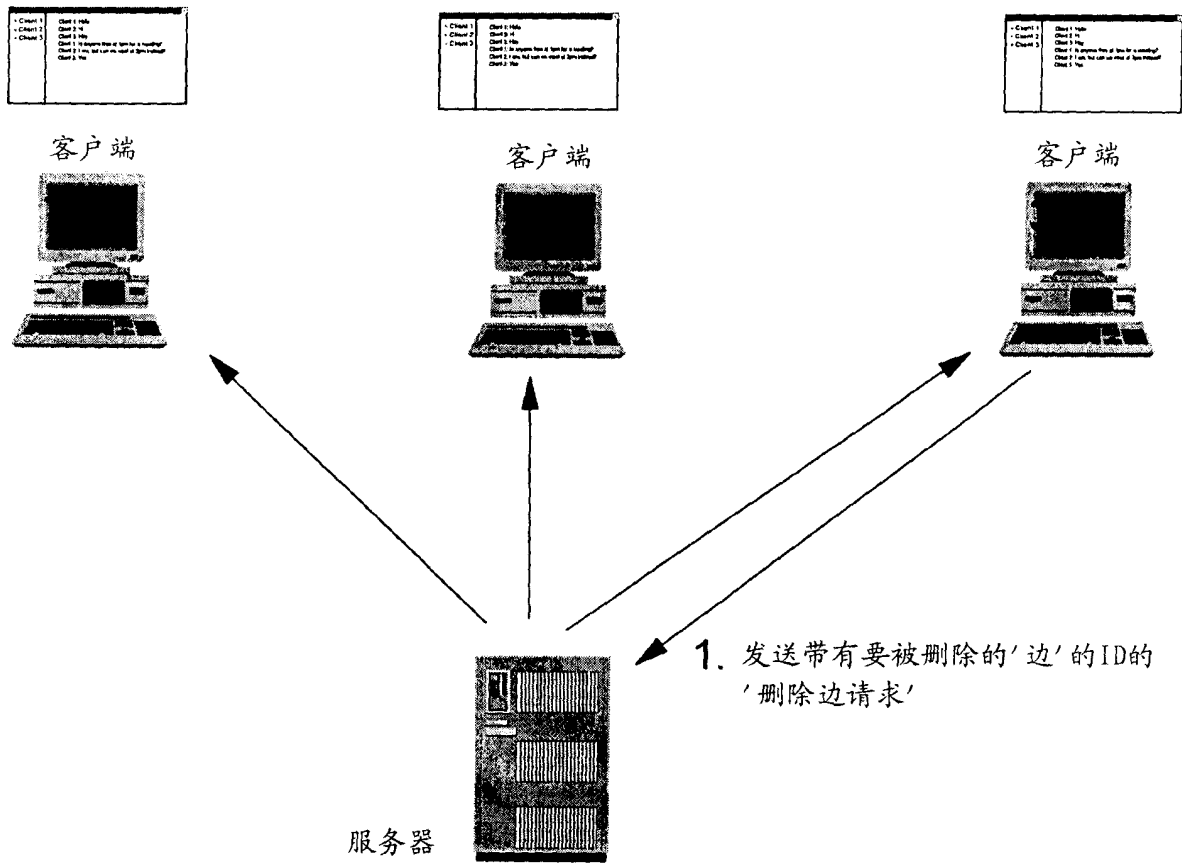


图 23

3. '删除边通知' 以及'边' ID, 被发送到该交谈中的所有客户端, 以便交谈客户端从屏幕上适当地去除'边'。



2. '边' 被从'边' 的集合中删除

图 24

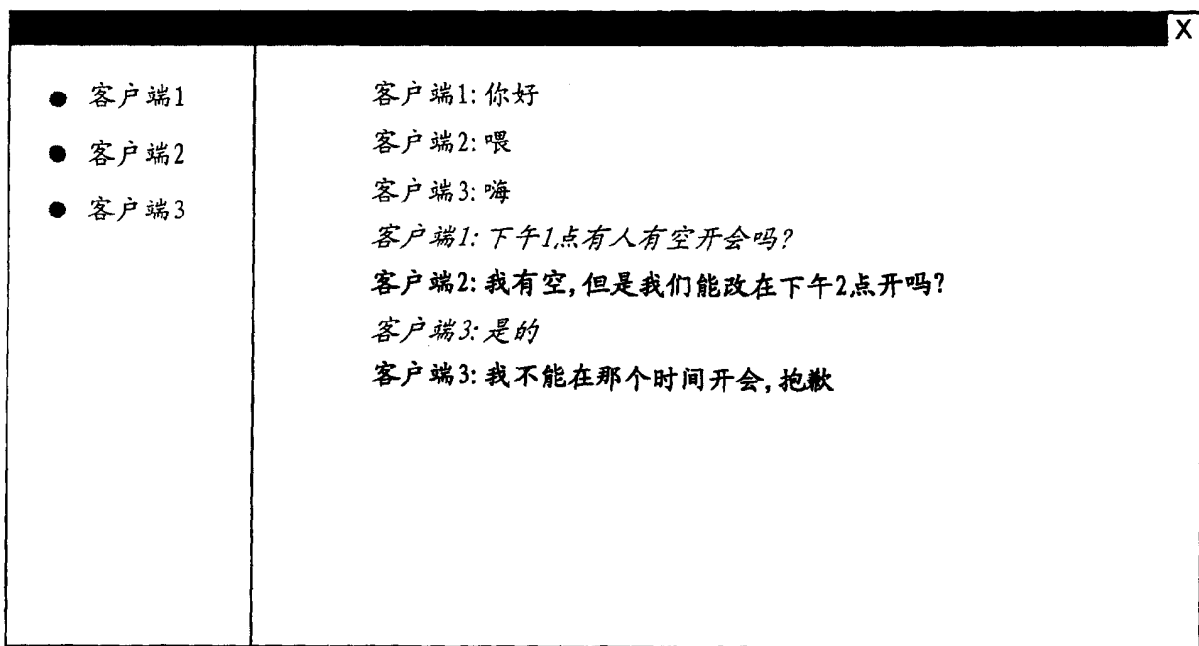


图 25

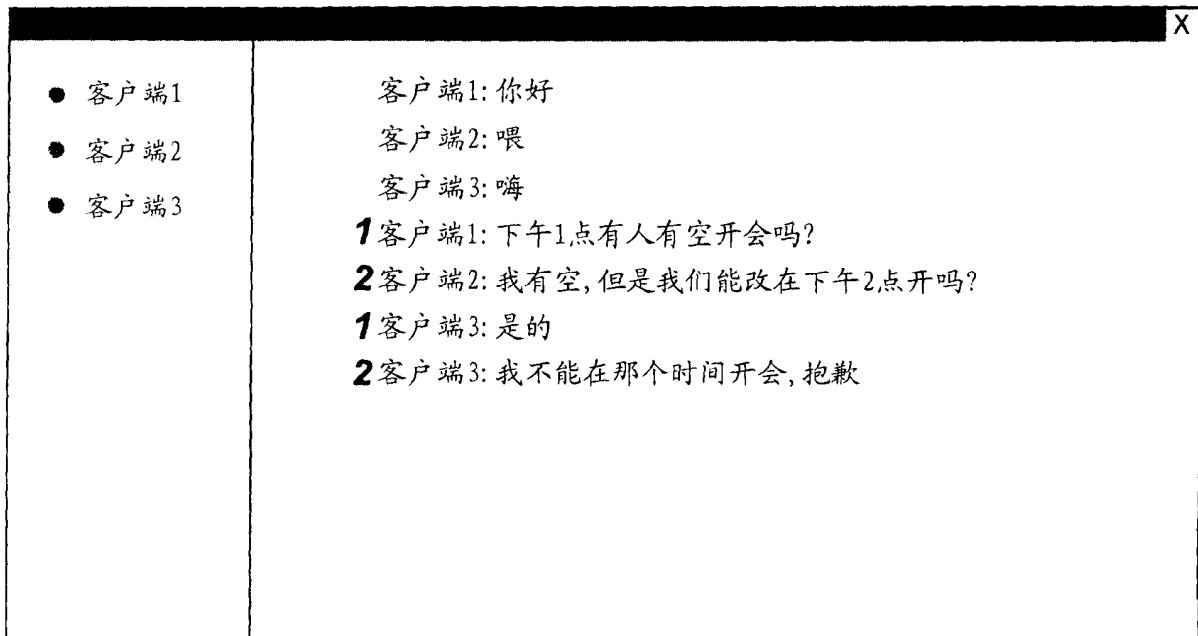


图 26

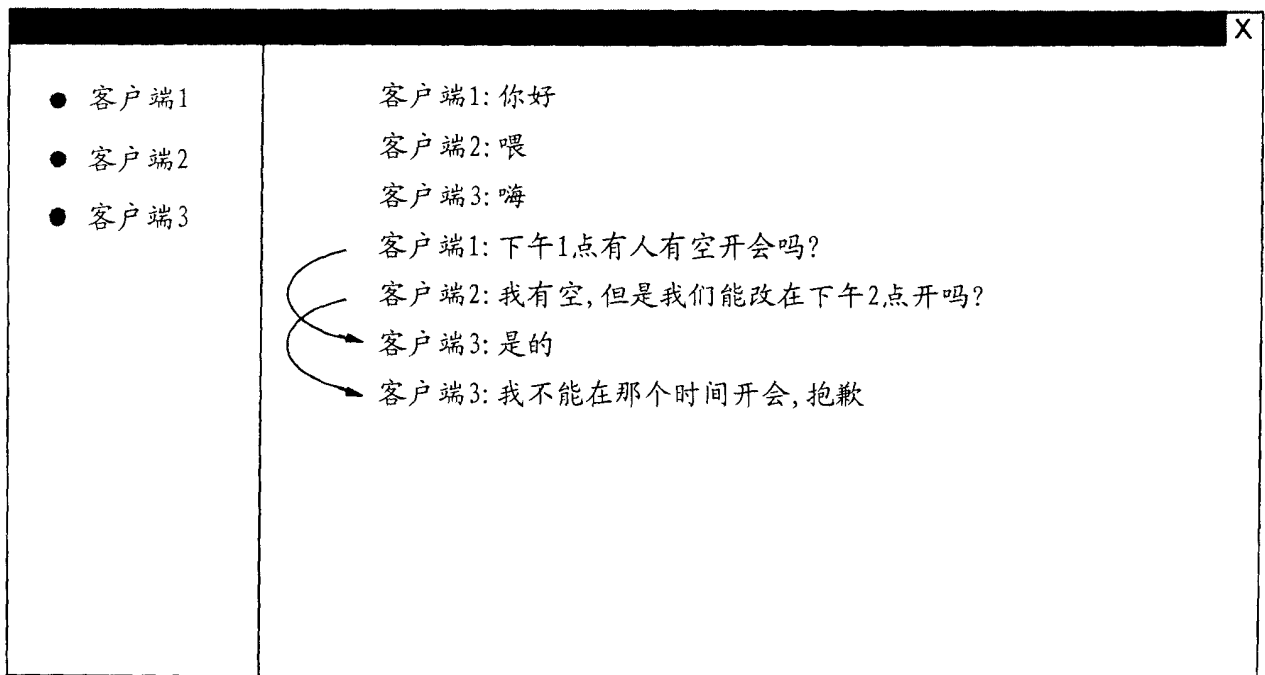


图 27