



# [12] 发明专利申请公布说明书

[21] 申请号 200580030695.7

[43] 公开日 2007年11月21日

[11] 公开号 CN 101076828A

[22] 申请日 2005.7.19

[21] 申请号 200580030695.7

[30] 优先权

[32] 2004.7.19 [33] US [31] 10/894,350

[86] 国际申请 PCT/US2005/025638 2005.7.19

[87] 国际公布 WO2006/014668 英 2006.2.9

[85] 进入国家阶段日期 2007.3.13

[71] 申请人 亚马逊科技公司

地址 美国内华达

[72] 发明人 维卡斯·古普塔 阿兰·H·弗穆兰

尤金·维 安德鲁·R·杰希

杰弗雷·P·贝佐斯

杜安·J·克劳斯

大卫·A·沙培尔

[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 康建忠

权利要求书 13 页 说明书 60 页 附图 18 页

## [54] 发明名称

执行自动授权的程序性交易

## [57] 摘要

描述了一种实现计算系统之间的交互的技术，比如通过第三方交易授权系统执行被自动授权的各方之间的交易。在一些情况下，交易为程序性交易，其涉及通过执行应用程序使用基于费用的 Web 服务，而交易授权系统按照预先由各方指定的专用授权指令授权和/或提供付款。授权指令可包括规定在其下潜在交易可被授权的条件预定指令规则集，而指令规则集均被相关的参考权标参考。在一个或多个潜在交易方向各方提供参考权标之后，根据与参考权标相关的指令规则集是否相容或被满足，交易授权系统可确定是否授权该交易。

由Web服务提供方ABC交互创建用户帐户的示例

欢迎, Web服务提供方ABC! 请在下面设置信息以创建您的用于WS付款处理系统("PHS")的帐户

1. 指定你的帐户名称和密码

名称:  密码:

可选的组织关系:

可选的证书:

2. 可选地指定你接收付款的一个或多个收支库  
你的PHS帐户(名为"MyPHSAccount")也可被用作收支库

库名称及类型	库号及截止日期	限制
<input type="text" value="ABC-Biz"/> <input type="text" value="bank account"/>	<input type="text" value="123456789-123-456789"/> <input type="text" value="&lt;none&gt;"/>	<input type="text" value="max_pay"/> <input type="text" value="\$50"/>
<input type="text" value="PerCCard"/> <input type="text" value="debit card"/>	<input type="text" value="1234-5678-1234-5678"/> <input type="text" value="12/05"/>	<input type="text" value="min_pay"/> <input type="text" value="\$0.05"/>
<input type="text" value="&lt;none&gt;"/>	<input type="text" value="&lt;none&gt;"/>	<input type="text" value="&lt;none&gt;"/>

3. 可选地指定你进行付款的一个或多个支付工具  
你的PHS帐户也可以被用作支付工具, 上限为\$1000.

库名称及类型	库号及截止日期	限制
<input type="text" value="ABC-Biz"/> <input type="text" value="bank account"/>	<input type="text" value="123456789-123-456789"/> <input type="text" value="&lt;none&gt;"/>	<input type="text" value="max_pay"/> <input type="text" value="\$50"/>
<input type="text" value="PerCCard"/> <input type="text" value="credit card"/>	<input type="text" value="1234-5678-5678-1234"/> <input type="text" value="01/05"/>	<input type="text" value="max_bal"/> <input type="text" value="\$2000"/>
<input type="text" value="&lt;none&gt;"/>	<input type="text" value="&lt;none&gt;"/>	<input type="text" value="max_pay"/> <input type="text" value="\$200"/>

1、一种 Web 服务提供方执行程序性交易的方法，该程序性交易基于 Web 服务提供方和其它各方的预定指令被自动授权，该方法在 Web 服务提供方的计算系统的控制下被执行，该方法包括：

从第一方接收对第一方和 Web 服务提供方之间的程序性交易的请求，所述交易包括 Web 服务提供方的至少一个 Web 服务功能的调用和对该 Web 服务功能从第一方到 Web 服务提供方的相关金融支付；

获取第一方的第一参考权标，其用于确定是否对程序性交易授权相关的金融支付，该第一参考权标与 Web 服务提供方不可访问的第一方的第一预定付款指令规则集相关；

将指示发送给第三方交易授权方，以自动确定是否授权从第一方到 Web 服务提供方的相关的金融支付，发送的指示包括获取的第一方的第一参考权标和与第一方不可访问的 Web 服务提供方的第二预定付款指令规则集相关的 Web 服务提供方的第二参考权标，交易授权方可访问第一和第二付款指令规则集，并且交易授权方的自动授权确定至少部分基于第一和第二付款指令规则集是否被确定为相容；并且

当确定相关的金融支付被授权时，将至少一个 Web 服务功能提供给第一方。

2、根据权利要求 1 的方法，包括当确定相关的金融支付被授权时，接收来自代表第一方的交易授权方的相关的金融支付。

3、根据权利要求 1 的方法，包括在确定相关的金融支付被授权之后，接收来自第一方的相关的金融支付。

4、根据权利要求 1 的方法，其中，请求的程序性交易还包括一个或多个附加的相关金融支付，所述金融支付分别由与交易相关的第一方、Web 服务提供方和第三方之一进行，并且分别针对第一方、Web 服务提供方、第三方和交易授权方中的不同的一个或多个，使得当一个或多个相应的付款指令规则集授权该附加付款时，附加的相关金融支付被各自进行。

5、根据权利要求1的方法，其中，接收的请求来自由第一方提供给不同的最终用户的应用程序，其中所述最终用户正执行该应用程序。

6、根据权利要求5的方法，其中，发送的指示还包括获取的与最终用户的第三预定付款指令规则集相关的最终用户的第三参考权标，并且其中确定检索的第一和第二付款指令规则集是否相容以便授权相关的金融支付，还包括确定第一和第二付款指令规则集是否与第三付款指令规则集相容。

7、根据权利要求1的方法，其中，接收的请求来自由另一方提供给第一方的应用程序，其中所述第一方正执行该应用程序。

8、根据权利要求1的方法，包括在接收请求之前，与交易授权方进行交互以定义第二付款指令规则集，并且作为响应接收来自交易授权方的第二参考权标。

9、根据权利要求8的方法，包括在由Web服务提供方并且在第一方的计算系统的控制下接收请求之前，与交易授权方进行交互以定义第一付款指令规则集，并且作为响应接收来自交易授权方的第一参考权标。

10、根据权利要求1的方法，其中权标为匿名的，使得Web服务提供方不能从第一权标获取有关第一付款指令规则集的信息，以及使得第一方不能从第二权标获取有关第二付款指令规则集的信息。

11、根据权利要求1的方法，其中第一权标为匿名的，使得Web服务提供方不能从第一权标获取有关第一方的信息。

12、根据权利要求1的方法，其中，向第三方交易授权方发送指示是基于由交易授权方提供的一个或多个Web服务的调用。

13、根据权利要求1的方法，其中，发送的指示进一步确定是否至少部分基于第一和第二付款指令规则集授权程序性交易。

14、根据权利要求1的方法，其中，发送的指示进一步确定在当前条件下是否每一第一和第二付款指令规则集均是有效的。

15、根据权利要求1的方法，其中，是否授权相关的金融支付的确定还包括由第一方和/或Web服务提供方按照第一和/或第二付款指

令规则集中指定的信息向交易授权方交互地提供信息。

16、根据权利要求 1 的方法，其中，第一和第二付款指令规则集均包括每一指定用于确定付款指令规则集是否授权程序性交易的标准的一个或多个规则。

17、根据权利要求 16 的方法，其中，确定第一和第二付款指令规则集相容以授权相关的金融支付包括确定每一第一和第二付款指令规则集的每一规则是否满足程序性交易相容。

18、根据权利要求 16 的方法，其中，每一付款指令规则集还包括规定付款指令规则集可授权交易的类型的一个或多个规则、规定付款指令规则集可授权的交易后修改条件的一个或多个规则、规定付款指令规则集可授权的要被接收或提供的付款类型的一个或多个规则、规定付款指令规则集可授权与之进行交易各方的类型的一个或多个规则。

19、根据权利要求 1 的方法，其中第一方和 Web 服务提供方分别还具有一个或多个使用指令规则集，每一使用指令规则集包括规定作为潜在程序性交易的一部分是否向另一方提供和/或从另一方接收信息访问的一个或多个规则。

20、根据权利要求 1 的方法，其中，第一方和 Web 服务提供方均具有其付款指令规则集与之相关的交易授权方的帐户，并且包括通过将第一方的帐户用作金融支付的支付工具并将 Web 服务提供方的帐户用作金融支付的收支库，代表第一方向 Web 服务提供方提供相关的金融支付。

21、根据权利要求 1 的方法，其中，第一和第二付款指令规则集被确定为对于程序性交易相容，并且包括，在从不同的第二方收到对在第二方和 Web 服务提供方之间的第二程序性交易的请求之后，其中所述交易包括 Web 服务提供方的 Web 服务功能的调用以及从第二方到 Web 服务提供方的第二相关金融支付，以及在基于第二付款指令规则集被确定为与同第二方的不同第三权标相关的第三付款指令规则集相容、确定第二程序性交易的第二金融支付被授权之后，将该 Web 服

务功能提供给第二方。

22、根据权利要求 1 的方法，其中，确定第一和第二付款指令规则集对于程序性交易相容，并且包括，在从第一方收到对在第一方和 Web 服务提供方之间的第二程序性交易的第二请求之后，以及在将第二指示发送给包括第一和第二参考权标的交易授权方之后，从交易授权方接收基于第一和第二付款指令规则集被确定为对于第二程序性交易不相容、确定不授权第二程序性交易的指示。

23、根据权利要求 1 的方法，其中，第一和第二付款指令规则集被确定为对于程序性交易不相容，并且包括，在收到用于确定是否对程序性交易授权相关的金融支付的第一方或 Web 服务提供方的第三参考权标的指示之后，基于与第三权标相关的第三付款指令规则集被确定为与第一和/或第二付款指令规则集相容，确定对程序性交易授权相关的金融支付。

24、根据权利要求 1 的方法，其中，每一第一和第二付款指令规则集均为 XML 文件。

25、根据权利要求 1 的方法，其中，获取第一参考权标包括随收到的请求接收第一参考权标。

26、一种计算机可读介质，通过执行以下方法其内容使计算系统在各方之间执行自动授权交易：

接收不同的第一方和第二方之间的潜在交易的指示，该交易包括作为交易的一部分的为提供一个或多个服务从第一方到第二方的相关的金融支付；

接收用于确定是否授权交易的第一方的第一参考和第二方的第二参考的指示；

从不同于第一和第二方的交易授权系统获取与第一参考相关的第一使用指令规则集是否被自动确定为与第二参考相关的第二使用指令规则集相容的指示，以这样的方式授权交易；并且

当第一和第二使用指令规则集被确定为相容时，执行交易的至少一部分。

27、根据权利要求 26 的计算机可读介质，其中，交易为程序性交易。

28、根据权利要求 27 的计算机可读介质，其中，程序性交易包括由第一和第二方之一调用由第一和第二方的另一方提供的 Web 服务。

29、根据权利要求 27 的计算机可读介质，其中，程序性交易包括第一和第二方之一对第一和第二方的另一方提供信息访问。

30、根据权利要求 27 至 29 之任一的计算机可读介质，其中，所述方法由第一方执行，并且其中程序性交易的至少一部分的执行包括将相关的金融支付从第一方提供到第二方，并且从第二方接收信息和/或功能作为提供一个或多个服务的一部分。

31、根据权利要求 27 至 29 之任一的计算机可读介质，其中，所述方法由第二方执行，并且其中程序性交易的至少一部分的执行包括从第一方接收相关的金融支付，并且将信息和/或功能提供给第一方和/或第三方作为执行一个或多个服务的一部分。

32、根据权利要求 27 至 29 之任一的计算机可读介质，其中，所述方法由第三方执行，并且其中程序性交易的至少一部分的执行包括从第二方接收信息和/或功能作为提供一个或多个服务的一部分。

33、根据权利要求 26 的计算机可读介质，其中，所述交易为程序性交易。

34、根据权利要求 26 的计算机可读介质，其中，所述交易为第一和第二方之间的物理交易，使得第一和第二方的至少一方将一个或多个物理服务提供给另一方。

35、根据权利要求 34 的计算机可读介质，其中，在确定第一和第二使用指令规则集相容之前完成提供一个或多个服务。

36、根据权利要求 34 的计算机可读介质，其中，至少部分基于确定第一和第二使用指令规则集相容执行提供一个或多个服务。

37、根据权利要求 26 的计算机可读介质，其中，确定第一和第二使用指令规则集相容包括评估第一和第二使用指令规则集的每一个中多个规则的每一个，以确定规则满足交易。

38、根据权利要求 26 的计算机可读介质，其中，交易还包括至少一个附加方，并且其中确定第一和第二使用指令规则集相容还包括确定第一和第二使用指令规则集与至少一个附加方的每一个的附加使用指令规则集相容。

39、根据权利要求 26 的计算机可读介质，其中，至少一个第一和第二参考为用于交易动态产生的参考权标，并且其相关的使用指令规则集至少部分基于为交易指定的指令。

40、根据权利要求 39 的计算机可读介质，其中，每一动态产生的参考权标为仅可用于交易的单次使用的权标。

41、根据权利要求 26 的计算机可读介质，其中，计算机可读介质为计算系统的内存。

42、根据权利要求 26 的计算机可读介质，其中，计算机可读介质为数据传输介质，用于传输生成的包含内容的数据信号。

43、根据权利要求 26 的计算机可读介质，其中，所述内容为当被执行时使计算系统执行所述方法的指令。

44、根据权利要求 26 的计算机可读介质，其中，所述内容包括用于自动授权各方之间的交易的一个或多个数据结构，该数据结构包含多个项，每项为与一方相关的使用指令规则集，并包含唯一相关的参考权标的指示和使用指令规则集的多个规则的指示。

45、一种配置成在各方之间执行自动授权的程序性交易的计算设备，包括：

交易请求方组件，被配置成在收到第一和第二方之间的程序性交易以及用于确定是否授权该程序性交易的第一方的第一参考权标和第二方的第二参考权标的一个或多个指示之后，获取与第一和第二参考权标相关的第一和第二付款指令规则集是否被满足以授权程序性交易的确定，所述程序性交易包括将来自第二方的一个或多个 Web 服务功能提供给第一方；以及

交易执行方组件，被配置成在确定第一和第二付款指令规则集相容之后，执行至少一部分程序性交易。

46、根据权利要求 45 的计算设备，其中，交易请求方组件和交易执行方组件在计算设备的内存中执行。

47、根据权利要求 45 的计算设备，其中交易请求方组件包含用于获取关于第一和第二付款指令规则集是否被满足以授权程序性交易的确定的装置，并且其中交易执行方组件包含用于执行至少一部分程序性交易的装置。

48、一种执行在 Web 服务提供方和消费方之间的程序性 Web 服务交易的计算机实现的方法，该程序性 Web 服务交易按照由各方提供的预定指令被自动授权给第三方交易授权方系统，所述方法包括，在 Web 服务提供方和消费方的计算系统的控制下：

对多个 Web 服务提供方的每一个，与第三方交易授权方系统交互，以定义 Web 服务提供方的一个或多个付款指令规则集并且作为响应接收每一付款指令规则集的相关的参考权标，每一付款指令规则集包括规定是否从 Web 服务消费方接受对交易的付款的多个规则，每一所述交易均使 Web 服务提供方基于费用提供至少一个 Web 服务，由交易授权方系统生成的每一参考权标以后作为潜在交易的一部分用于参考相关的付款指令规则集；

对多个 Web 服务消费方的每一个，与第三方交易授权方系统交互，以定义 Web 服务消费方的一个或多个付款指令规则集并且作为响应接收每一付款指令规则集的相关的参考权标，每一付款指令规则集包括规定是否向 Web 服务提供方提供对交易的付款的多个规则，每一所述交易均使 Web 服务消费方接收基于费用提供的至少一个 Web 服务，由交易授权方系统生成的每一参考权标以后作为潜在交易的一部分用于参考相关的付款指令规则集；并且

使用生成的参考权标和相关的付款指令规则集以允许对涉及 Web 服务的交易进行付款，

对多个 Web 服务消费方的每一个，识别每一个由 Web 服务提供方之一基于相关费用提供的一个或多个 Web 服务，并且程序性地调用来自提供该 Web 服务的 Web 服务提供方的每一识别的 Web 服务，每



一 Web 服务的程序性调用包括提供 Web 服务消费方的参考权标之一；并且

对多个 Web 服务提供方的每一个，接收一个或多个调用，该调用均针对由 Web 服务提供方提供的 Web 服务并且均包括用于 Web 服务消费方进行调用的参考权标，并且每一收到的对 Web 服务的调用，

向交易授权方系统发送请求，以授权潜在的交易，该潜在的交易包括 Web 服务提供方将调用的 Web 服务提供给 Web 服务消费方，以使基于费用的调用与调用的 Web 服务相关，发送的请求包括随接收的调用包含的 Web 服务消费方的参考权标，以及 Web 服务提供方的参考权标之一；并且

当交易授权方系统基于与包括在发送请求中的参考权标相关的付款指令规则集相容确定潜在的交易被授权时，通过交易授权方系统从 Web 服务消费方接收调用的 Web 服务的相关费用，并将调用的 Web 服务提供给 Web 服务消费方，使得 Web 服务提供方和消费方可根据由相关的参考权标指示的预定指令进行包括对 Web 服务付款的交易。

49、根据权利要求 48 的方法，包括在交易授权方系统的控制下，响应每一个请求在 Web 服务提供方和 Web 服务消费方之间授权潜在交易的多个接收的请求的每一个，确定是否授权潜在交易，其中所述交易包括与要提供的 Web 服务相关的费用，每一授权请求包括 Web 服务提供方的参考权标和 Web 服务消费方的参考权标，所述确定是否授权潜在交易通过以下执行：

检索与包括在授权请求中的参考权标相关的付款指令规则集；

确定检索的付款指令规则集是否相容，使得 Web 服务提供方的付款指令规则集中的规则允许以 Web 服务消费方的付款指令规则集中指定的方式接受来自 Web 服务消费方的相关费用，并使得 Web 服务消费方的付款指令规则集中的规则允许以 Web 服务提供方的付款指令规则集中指定的方式将相关的费用支付给 Web 服务提供方；并且

当确定检索的付款指令规则集相容时，授权潜在的交易，并以

Web 服务提供方的付款指令规则集中指定的方式将相关的费用提供给 Web 服务提供方，并且以 Web 服务消费方的付款指令规则集中指定的方式从 Web 服务消费方获取付款。

50、根据权利要求 48 的方法，其中，Web 服务消费方为应用开发方，其向不同的最终用户提供包括用于 Web 服务消费应用开发方的一个或多个参考权标的应用程序，并且其中 Web 服务的程序性调用均来自正由最终用户执行的应用程序，并将包括的参考权标之一用于提供应用程序的 Web 服务消费应用开发方。

51、根据权利要求 50 的方法，其中，来自正由最终用户执行的应用程序的 Web 服务的每一程序性调用还包括提供最终用户的不同的参考权标，该参考权标与最终用户的付款指令规则集相关，所述用户的付款指令规则集包括规定是否对均包括接收至少一个 Web 服务的交易提供付款的规则，并且其中每一由 Web 服务提供方发送给交易授权方系统的请求还包括最终用户的参考权标，该参考权标随接收的对 Web 服务的调用被包含，使得交易授权方系统进行的潜在交易被授权的确定还基于与 Web 服务提供方、Web 服务消费方和最终用户的参考权标相关的付款指令规则集的相容性。

52、根据权利要求 48 的方法，其中，确定潜在交易的付款指令规则集相容包括：

对与 Web 服务提供方的参考权标相关的付款指令规则集中的每一规则，确定规则被 Web 服务消费方、与 Web 服务消费方的参考权标相关的付款指令规则集、关于潜在交易的信息、以及执行与潜在交易有关的应用程序的最终用户的一个或多个满足；以及

对与 Web 服务消费方的参考权标相关的付款指令规则集中的每一规则，确定规则被 Web 服务提供方、与 Web 服务提供方的参考权标相关的付款指令规则集、关于潜在交易的信息、以及执行与潜在交易有关的应用程序的最终用户的一个或多个满足。

53、根据权利要求 52 的方法，其中，每一付款指令规则集还包括规定付款指令规则集可授权的交易类型的一个或多个规则、规定付款

指令规则集可授权的交易后修改条件的一个或多个规则、规定付款指令规则集可授权的要被接收或提供的付款类型的一个或多个规则、规定付款指令规则集可授权以与之进行交易的各方的类型的一个或多个规则。

54、根据权利要求 48 的方法，其中，每一 Web 服务提供方和 Web 服务消费方具有有不同的规则的多个付款指令规则集和不同的相关参考权标，使得 Web 服务提供方和 Web 服务消费方之间的每一潜在交易包括由每一 Web 服务提供方和消费方选择其付款指令规则集之一，用于确定是否授权潜在交易。

55、根据权利要求 48 的方法，其中，每一 Web 服务提供方和 Web 服务消费方具有交易授权系统的帐户，其付款指令规则集与该交易授权系统相关，其中 Web 服务消费方的帐户可被用作付款给 Web 服务提供方的支付工具，并且其中 Web 服务提供方的帐户可被用作从 Web 服务消费方付款的收支库，使得对至少一些潜在交易将相关的费用从 Web 服务消费方提供给 Web 服务提供方涉及使用在其交易中涉及的 Web 服务提供方和消费方的帐户。

56、根据权利要求 48 的方法，其中，交易授权方系统还对向 Web 服务提供方和消费方提供生成的参考权标以以后使用的一个或多个中的每一个收费，确定潜在交易的付款指令规则集是否相容，并且针对潜在交易将相关的费用从 Web 服务消费方提供给 Web 服务提供方。

57、根据权利要求 48 的方法，其中，参考权标为匿名的，使得潜在交易方不能获取关于与潜在交易的另一方的参考权标相关的付款指令规则集的信息，并且不能从参考权标获取关于另一方的信息。

58、根据权利要求 48 的方法，其中，作为潜在交易的一部分由 Web 服务提供方将调用的 Web 服务提供给 Web 服务消费方包括将响应消息提供给 Web 服务消费方，该响应消息包括与由 Web 服务提供方执行 Web 服务有关的信息，并且包括当交易授权方系统确定随后潜在交易未被授权时，将与 Web 服务有关的错误消息从 Web 服务提供方提供给 Web 服务消费方。

59、一种用于自动执行涉及用户的授权活动的计算机实现的方法，包括：

接收涉及第一用户和不同的第二实体的潜在活动的指示；

接收用于自动确定是否授权指示的活动的第一用户的第一参考和第二实体的第二参考的指示；

从不同于第一用户和第二实体的交易授权系统获取与第一参考相关的第一使用指令规则集是否被自动确定为同与第二参考相关的第二使用指令规则集相容的指示，以这样的方式授权指示的活动；并且

当第一和第二使用指令规则集被确定为相容时，实现指示的活动的执行。

60、根据权利要求 59 的方法，其中，指示的活动为在第一用户和第二实体之间提出的交易，其包括作为交易的一部分对提供一个或多个服务从第一方到第二方的相关的金融支付，并且其中实现指示的活动的执行包括执行至少一部分交易。

61、根据权利要求 59 的方法，其中，指示的活动包括第一用户获取对可从第二实体得到的受限制信息的访问。

62、根据权利要求 59 的方法，其中，指示的活动包括第一用户获取对可从第二实体得到的受限制服务的访问。

63、根据权利要求 59 的方法，其中，指示的活动包括第一用户与第二实体达成一个或多个指示的合同条款。

64、根据权利要求 63 的方法，其中，一个或多个指示的合同条款包括用于第一用户和第二实体之间的一个或多个交易的争端解决策略。

65、根据权利要求 63 的方法，其中，一个或多个指示的合同条款包括关于在第一用户和第二实体之间提供付款的退款的至少一个条款。

66、根据权利要求 59 至 65 之任一的方法，其中，第二实体为不同于第一用户的第二用户。

67、一种用于自动执行各方之间的授权交易的计算机实现的方法，

包括:

接收在不同的第一方和第二方之间的交易的指示, 该交易包括提供一个或多个服务;

接收用于确定是否授权该交易的第一方的第一参考和第二方的第二参考的指示, 第一参考以保护与第一方相关的保密信息的方式被生成;

获取与第一参考相关的第一方的第一预定授权指令是否被自动确定为与第二方的第二预定授权指令相容, 以这样的方式授权交易; 并且

当第一和第二使用指令规则集被确定为相容时, 以保护与第一方相关的保密信息的方式执行至少一部分交易。

68、根据权利要求 67 的方法, 其中, 第一方的预定的授权指令包括具有多个规则的指令规则集。

69、根据权利要求 67 和 68 之任一的方法, 其中, 保护与第一方相关的保密信息包括防止第二方获取该保密信息。

70、根据权利要求 67 至 69 之任一的方法, 其中, 第一参考以这样的方式被生成, 即保护的与第一方相关的保密信息包括第一方的身份的指示。

71、根据权利要求 67 至 69 之任一的方法, 其中, 第一参考以这样的方式被生成, 即保护的与第一方相关的保密信息包括第一方的预定的授权指令。

72、一种用于自动执行各方之间的授权交易的计算机实现的方法, 包括:

接收在不同的第一方和第二方之间的交易的指示, 该交易包括提供一个或多个服务;

接收用于确定是否授权该交易的第一方的第一参考和第二方的第二参考的指示;

获取与第一参考相关的第一使用指令规则集是否被自动确定为与第二参考相关的第二使用指令规则集相容的指示, 以这样的方式授权

交易；并且

当第一和第二使用指令规则集被确定为相容时，提供交易被授权的指示并提供与交易执行有关的功能的一个或多个类型的指示。

73、根据权利要求 72 的方法，还包括，当第一和第二使用指令规则集被确定为相容时，执行至少一部分交易并提供实现交易的执行的一个或多个类型的辅助功能。

74、根据权利要求 72 至 73 之任一的方法，还包括，当第一和第二使用指令规则集被确定为相容时，向至少第一和第二方之一提供对应于交易的付款。

75、根据权利要求 72 至 74 之任一的方法，还包括，当第一和第二使用指令规则集被确定为相容时，从至少第一和第二方之一获取对应于交易的付款。

76、根据权利要求 72 至 75 之任一的方法，其中，与交易执行有关的一个或多个类型的功能包括保证要对交易提供付款的保险服务。

77、根据权利要求 76 的方法，其中，保险服务的使用使预期的付款接受方能够以基本实时的方式获得对付款的存取。

78、根据权利要求 77 的方法，其中，保险服务使预期的付款接受方能够在从付款的预期支付方获取付款之前获得对付款的存取。

79、根据权利要求 77 至 78 之任一的方法，其中，保险服务使预期的付款接受方能够在付款支付方可能对付款质疑期间获得对付款的存取。

80、根据权利要求 72 至 79 之任一的方法，其中，与交易性能有关的一个或多个类型的功能包括保证要对交易提供付款的保险服务，以便确保付款被提供。

81、根据权利要求 80 的方法，其中，保险服务的使用使预期的付款接受方能够以基本实时的方式获得对付款的存取。

## 执行自动授权的程序性交易

### 技术领域

下面的公开一般涉及便于在计算系统之间交互，更具体地说，涉及根据各方的预定授权指令在被自动授权的各方之间执行交易，比如在各方的执行应用程序之间授权基于费用的 Web 服务程序性 (programmatic) 交易。

### 背景技术

存在各种机制以使计算机和执行软件应用能够进行程序性交互。例如，早已存在的远程过程调用 (“RPC”) 协议允许一个计算机上的程序使另一计算机上的程序被执行，并且各种面向对象的体系结构比如 CORBA (“公用对象请求代理体系结构”) 和 DCOM (“分布式对象组件模型”) 也提供类似的功能。此外，已实现了各种中间程序，用来连接单独的应用 (经常为不同类型并来自于不相关的源) 以允许通信。例如，存在各种 EDI (“电子数据交换”) 网络，其提供标准机制，以使网络的一个用户的计算机系统能够向网络的另一用户的计算机系统发送数据。

万维网 (“Web”) 的广泛普及给计算机交互通信提供了附加的机会。例如，许多当前的 Web 应用涉及用户从 Web 服务器交互地请求网页 (例如，通过执行用户的 Web 浏览器应用)，并且作为响应接收请求的信息。除这种请求信息的交互用户规范之外，还存在不断增加的 Web 的使用，用来支持远程应用的程序性交互，以通过定义的 API (“应用程序接口”) 比如通过 Web 服务交换信息。

Web 服务使得不同种类的应用和计算机能够进行交互，并可利用各种底层协议和技术来定义和实现。例如，响应指定为 URI (“统一资源标识符”) 比如包括指定的操作和一个或多个查询参数的 URL (“统

一资源定位符”)的 Web 服务调用请求,一些 Web 服务实现使用 HTTP (“超文本传输协议”)以 XML (“可扩展标记语言”)格式返回数据。在其它实现中,附加的底层协议被用于各种用途,比如用于标准消息交换的 SOAP (“简单对象访问协议”)、用于描述服务调用的 WSDL (“Web 服务描述语言”)、用于发现可用服务的 UDDI (“通用描述、发现和集成服务”)。

尽管 Web 服务使得各种应用和计算机能够进行交互,但当前的实现和 Web 服务的使用具有各种限制。例如,虽然一些 Web 服务的提供方愿意使他们的 Web 服务可免费使用,但多数可用的和可靠的 Web 服务经常不是免费使用的。不幸地,对于 Web 服务的提供方和这些 Web 服务的潜在消费方来说,商定付费和交易的其它条款的机制可能是费时并难以使用的。而且,可能经常有阻碍这种商定的安全和信任的问题,使得潜在消费方可能不能充分保证未来的 Web 服务提供方会可靠地并准确地提供希望的 Web 服务,而 Web 服务提供方可能不能保证他们的 Web 服务仅由授权的消费方使用并且仅按照由提供方指定的任意条件使用。此外,应用开发方可能不愿意使他们的应用程序调用基于费用的 Web 服务,如果他们不能限制并控制由将进行这些 Web 服务的调用的应用程序的最终用户使用这样的 Web 服务的话。最后,通常不存在易用的机制,使 Web 服务提供方能够对由消费方使用他们的 Web 服务收取指定的费用。

因此,提供一种致力于与 Web 服务的使用相关的这些和其它问题和/或其它的便于计算机系统和执行应用的交互的解决方案将是有益的。

#### 附图说明

图 1A 至 1E 示出定义在授权程序性交易中所使用的的使用指令规则集的示例。

图 2 是示出 Web 服务付费处理系统的示例实施例的构成和信息流的网络图。



图 3 是示出适于执行付费处理系统的实施例的计算系统的实施例的框图。

图 4 是用户账户管理例程的实施例的流程图。

图 5 是交易授权和处理例程的实施例的流程图。

图 6 是交易确认子例程的实施例的流程图。

图 7A 和 7B 是执行付费处理交易子例程的实施例的流程图。

图 8A 至 8C、9A 至 9B 和 10A 至 10C 示出交易各方之间的各类交互的示例。

### 具体实施方式

下面，例如通过在一些实施例中提供第三方交易授权系统，描述便于在计算系统之间进行交互的软件设施，所述第三方交易系统根据由各方预先指定的专用授权指令自动地授权在各方之间的交易和/或该交易的金融支付。在一些实施例中，一些或所有交易为在执行计算机程序之间进行的程序性交易，比如通过与 Web 服务消费方相关的应用程序涉及使用 Web 服务提供方的 Web 服务的交易。此外，在一些实施例中，交易授权系统还以其它各方的一个或多个的名义向各方的一个或多个提供授权的付费。交易授权系统使得交易各方能够自动确定交易是否被授权，如果被授权，则继续进行交易的执行。

交易授权系统可被用于对各方之间的各种类型的交互授权交易和/或付款，而图 8A 至 8C 示出了一些这样的交互的示例。特别是，图 8A 示出两方交易 830a 的示例，其中，一方 805a 请求要由一方 810a 提供的服务，而提供服务具有相关的费用。交易授权系统 820a 使用有关各方 805a 和 810a 的信息（例如从向交易授权系统进行程序性调用请求授权的各方之一接收的信息），以便确定是否授权从服务请求方到服务请求方的交易和/或提供付款 850a，以支付与交易相关的费用。交易和/或付款被授权之后，交易授权系统提供授权的指示（例如，作为授权请求调用的响应），并且在一些实施例中还可以以服务请求方的名义实际进行授权的付款。下面将更详细地描述关于授权处理 840a

如何发生的细节。

说明的授权和付款技术可被用于各种情况。例如，提供的服务可涉及两方之间的程序性交易（例如，服务请求方调用服务提供方提供的 Web 服务），或可涉及其它类型的交易（例如，各方之间的物理交易，比如将物理产品或服务从一方提供到其它方）。交易授权系统的授权交互可以以各种方式类似地被执行，比如在交易期间（例如，使交易不被完成，除非收到适当的授权请求），或者以独立于交易的方式（例如，在完成交易之后，以便向服务提供方提供付款）。此外，在执行授权中所使用的关于各方的信息可类似地具有各种形式，如下面更详细描述。

图 8B 提供了对交易 830b 的更复杂付款 850b 的示例，其中，一方 805b 请求要由一方 810b 提供的服务。在该示例中，在交易授权方 820b 授权交易（未示出）之后，部分付款可被提供给服务提供方 810b 和交易授权方 820b（例如，基于交易授权方对执行交易和/或付款的授权收取费用），并且部分付款可由服务请求方和一个或多个其它付款提供方（例如，提供正由服务请求方执行的应用程序的应用开发方）来提供。在其它情况下，各方之间的付款可以以其它方式被提供，比如，如果付款提供方提供所有付款，如果付款提供方将部分应付款提供给服务提供方和交易授权方之一，并且服务请求方将部分应付款提供给其它服务提供方和交易授权方等。

图 8C 提供具有多个参与方 860a 至 860n 的更复杂交易 830c 的示例。在该示例中，一些或所有这些各方以及一个或多个其它付款提供方 815c 可提供信息给交易授权方 820c 或交互，以使交易授权方能够确定交易和/或相关的金融支付是否被授权。在授权处理 840c 提供批准之后，可由一个或多个参与方和/或付款提供方对其它的参与方和/或交易授权方进行付款（未示出）。

如上所述，当确定是否授权交易时，交易授权系统可将有关各方的各种信息用于一个交易。具体地说，作为这样的交易的潜在方的用户可利用交易授权系统首先定义以后用于授权交易的一个或多个类型

的使用指令规则集，比如用于授权基于费用的交易和/或这种交易的相关的金融支付的付款指令规则集。一方的每一这样的付款指令规则集可包括规定条件的一个或多个指定的规则，在所述条件下，付款指令规则集可对该方的潜在交易和/或其相关的金融支付授权，如下面更详细所述。

例如，潜在的 Web 服务消费方（例如，希望他们的应用程序能够在指定的环境下调用 Web 服务的应用开发方）可指定付款指令规则集的规则，该规则集限定可被该付款指令集授权的交易的 Web 服务消费方的金融风险（例如，付款指令规则集可授权交易的次数、付款指令规则集可授权的每一交易的最大付款量、付款指令规则集可授权的最大总付款量、付款指令规则集的截止日期等）。类似地，潜在的 Web 服务提供方可指定付款指令规则集的规则，该规则集限定可被该付款指令规则集授权的交易的 Web 服务提供方的责任（例如，限定可购买访问 Web 服务的消费方、限定可被用于购买访问 Web 服务的付款类型、限定对他们使用的 Web 服务不满意的消费方的有效补偿等）。此外，潜在的 Web 服务呼叫方（例如，由调用一个或多个 Web 服务的应用开发方提供的应用程序的最终用户）可指定付款指令规则集的规则，该规则集限定可被该付款指令规则集授权的交易的呼叫方的责任（例如，限定呼叫方被授权启动的程序性交易的类型和/或可进行购买访问程序性服务的付款的类型，比如对服务提供方的付款，和/或付给交易授权服务和/或其它第三方服务比如付款服务的费用）。

收到用户的指定的使用指令规则集之后，可选择地，在首先准许指令规则集之后（例如，基于指令规则集是否包括所需信息（如果有的话）的验证，和/或是否处于正确的形式的验证），交易授权系统以与用户相关的方式存储指令规则集（例如，将指令规则集与交易授权系统的用户的帐户相关）。交易授权系统还生成参考了指令规则集的参考权标（reference token），将参考权标与指令规则集相关（例如，通过与指令规则集一起存储参考权标的指示），并且将参考权标提供给用户，用于以后用于引用指令规则集。如其它更详细所述，参考权

标可以各种方式被生成，并可采用各种形式（例如，保证为唯一的长随机数），并且在一些实施例中，对单一指令规则集可生成多个参考权标，多个参考权标可与单一指令规则集相关。在一些实施例中，参考权标以允许用户匿名及/或专用地使用参考权标这样的方式被生成，比如通过缺少与用户相关的任何识别信息和/或通过缺少有关与参考权标相关的指令规则集的条件任何信息（例如，为了防止涉及这种参考权标的潜在交易的其它各方获取有关用户和/或相关的指令规则集的信息）。

在限定使用指令规则集并生成相关的参考权标之后，随后，它们可按照指令规则集的先前提定的条件被用于允许程序性交易的自动授权。特别是，在一些实施例中，潜在程序性交易的两方或多方（例如，Web 服务提供方、Web 服务消费方和任选的最终用户呼叫方）均提供用于授权潜在交易的参考权标，以及有关潜在交易的各种信息。然后，这些参考权标和有关潜在交易的各种信息可被提供到交易授权系统，用于交易是否被授权的自动确定（例如，比如以实时方式的动态确定）。如下面更详细所述，在一些实施例中，潜在交易是否根据提供的参考权标被授权的交易授权系统的自动确定包括检索与这些参考权标相关的指令规则集，然后确定指令规则集是否相容（compatible）。

关于被提供给交易授权系统的潜在交易的信息可在各种实施例中和在各种情况下变化。例如，在一些情况下，关于交易的信息可被限定为仅为要由一方支付给另一方的交易量，而在其它情况下，可提供各种关于交易的附加细节（例如，交易的类型）。此外，在一些实施例中，交易授权系统的调用还可指定关于潜在交易的两方或多方的每一方需要支付什么（如果有的话），是否支付给另一方并/或作为费用给交易授权系统的细节。

使用指令规则集是否相容的确定可在各种实施例中以各种方式被执行，比如通过确定是否每一指令规则集中的每一规则均适于当前的潜在交易。这种规则满意度可针对不同类型的规则以各种方式出现，比如基于关于潜在交易的信息（例如，交易的类型和/或交易的相关付

款量)、关于交易的一个或多个其它方的信息(例如,各方是否满意指定的规则)、关于对潜在交易指定的一个或多个其它的指令规则集的信息(例如,其它指令规则集中的一个或多个对应的规则是匹配或是以其它方式适于规则)等。相反地,在一些情况下,指令规则集可被确定为不相容,因为根据潜在交易时的条件,比如如果指令规则集不是当前可用的(例如,其已经到期或在其它方面不适用于当前使用,比如基于允许使用的总数或已超出的总允许付款量)及/或如果其不能被用于潜在的交易,则一个或多个指令规则集(以及其相关的参考权标)对当前的潜在交易是无效的。

当具有一个或多个相关的金融支付的潜在交易被自动确定将被授权时,在一些实施例中,交易授权系统还可在各方之间适当地执行或实现付款交易。例如,在一些实施例中,付款提供用户可具有事先指定的一个或多个支付工具(instrument)(例如,信用卡、银行帐户等),如果有的话,可以以适当的方式从一个或多个这些支付工具中获取金融支付(例如,从付款提供用户的付款指令规则集所指定的支付工具中)。类似地,付款接收用户可具有事先指定的一个或多个收支库(payment repository)(例如,银行帐户、借记卡等),如果有的话,可以以适当的方式将金融支付提供给一个或多个这些收支库(例如,提供给由付款接收用户的付款指令规则集指定的收支库)。

在交易授权系统授权潜在的交易并且对潜在的交易可选地执行任何相关的金融支付之后,交易的各方可随后以适当方式执行交易。例如,当交易为基于费用的Web服务交易时,Web服务消费方可调用该Web服务,并且Web服务提供方可将Web服务的相关功能提供给消费方。

在一些实施例中,可由交易授权系统施加附加限制,即使潜在交易被授权和/或被执行之后。例如,在一些情况下,可不允许付款接收方马上提取或接受这些付款,比如通过限制访问其中已存入付款的交易授权系统的帐户和/或通过一些或所有收到的付款延迟存入接收方的另一收支库。这种付款访问限制可针对各种原因并以各种方式被

执行，包括在交易被授权和/或被执行之后的指定时间段期间使任何纠纷能被解决以及使任何退款能被自动进行（例如，付款接收方退还应退款量）。在一些实施例中，附加的相关功能还可由交易授权系统或第三方提供，比如基于费用的担保或保险服务，其保证或提供任何这样的退款的资金，从而使得付款接收方能够立即或更快捷地存取任何收到的付款。

在一些实施例中，交易授权系统的用户可更一般地定义用于除授权基于费用的交易之外的情况下的其它类型的使用指令规则集，不管是代替还是增加付款指令规则集。例如，这样的使用指令规则集均可包括一个或多个指定的规则，其规定使用指令规则集可授权不包括费用的潜在交易下的条件，比如确定是否提供对受限制信息或受限制服务的访问。

此外，使用指令规则集中的规则可在各种实施例中以各种方式被指定和使用，并在全部实施例中可具有各种形式。例如，在一些实施例中，规则可指定验证信息的类型、对于该信息执行的评估的类型、和用于确定评估结果是否满足规则的测试。更普遍地，在其它实施例中，规则可反映可被验证和/或评估的任何条件或信息，以确定其在指定的情况下是否被满足。例如，在一些实施例中，规则可被指定使用 XML 格式，并且在一些实施例中，指令规则集还可被指定和/存储为 XML 文档。

此外，在一些实施例中，在创建使用指令规则集时用户可以各种预定类型的规则中选择，比如每一预定类型的规则指定与使用指令规则集有关的条件类型，以及每一用户选择包括指定用来定制预定的规则类型的一个或多个参数值，以便影响其使用条件如何或何时满足正创建的指令规则集。此外，在一些实施例中，用户可指定各种用户特定规则，比如指示用于确定规则何时被满足的用户指定的过程的规则（例如，通过提供当被调用时提供关于规则是否被满足的指示的规则的可执行代码，或者通过指示一种方式以自动获取指示规则是否被满足的响应，比如通过调用可访问的用户指定的 Web 服务或使用指定的

参数值的其它功能)。

在一些实施例中, 规则的评估还可涉及获取要被用作关于规则是否被满足的部分确定的附加信息, 比如以自动方式和/或以对于一个或多个用户的交互方式(例如, 通过询问创建指令规则集的用户以确认评估和/或提供附加信息)。此外, 在一些实施例中, 其中, 提供涉及使用指令规则集的安全和/或保密的功能, 指令规则集中的一个或多个规则可被用来提供该功能(例如, 通过限定谁能访问和/或修改指令规则集), 而在其它的实施例中, 可以以其它方式提供这样的功能(例如, 通过控制访问一个或多个指令规则集可与其相关的用户帐户)。

此外, 付款/使用指令规则集和其规则可在各种实施例中以各种方式被创建, 比如交互地通过由交易授权系统提供的图形用户接口(“GUI”), 和/或通过调用针对该目的的由交易授权系统提供的一个或多个 Web 服务。可以以各种方式实现这样的 Web 服务, 比如使用基于 HTTP 上的 XML 的 API(例如作为集中于资源的分布交互模型的代表性状态传输(或“REST”)的一部分)、和/或基于 SOAP 的使用和/或与 Web 服务相关的其它协议。此外, 在指令规则集被创建并与参考权标相关之后, 在一些实施例中, 可不允许修改指令规则集。或者, 在一些实施例中, 可修改这样的指令规则集, 并且将为新的指令规则集生成针对修改的指令规则集的新的参考权标(例如用来代替先前的参考权标)。而且, 在一些实施例中, 参考权标和/或相关的指令规则集可在想要使用时被动态地创建(例如, 作为试图调用程序性服务和/或向第三方授权方发送授权请求的一部分或在其之前), 作为一个示例, 在一些实施例中, 可创建并使用一次(one-time)或单次使用(single-use)权标和/或指令规则集(例如, 用于特定的潜在交易)。

此外, 在一些实施例中, 其中, 基于费用的交易被授权并且金融支付被处理, 一些或所有用户还可拥有一个易于该支付的交易授权系统的帐户, 比如可被用户用作支付工具和/或收支库的帐户。在一些实施例中, 这种帐户和/或其它金融工具/库可提供很少量的微付款。此外, 在一些实施例中, 交易授权系统还可提供要被提供和/或接收(例如定

期地，比如一个月)的金融支付的集合，以便将与金融支付交易相关的费用最小化和/或能达到最小付款量。对于其它类型的帐户，在一些实施例中，这种用户帐户可具有与其使用相关的各种条件和限制，比如允许信贷结算的最大值，以及各种安全和访问控制(例如基于密码)。

此外，虽然交易授权系统在上面已被描述为在各种实施例中提供各种功能和功用(capability)，但在一些实施例中，一些这样的功能和功用可分解至一个或多个系统。例如，在一些实施例中，一些或所有用户可能已有另一系统的用户帐户，交易授权系统附属于该系统或交易授权系统访问该系统，如果这样的话，一些或所有与帐户有关的功能可由该其它系统连同交易授权系统一起来提供(例如，存储指令规则集、生成参考权标、提供收支库/支付工具功能等)。

出于例证性目的，下面描述一些软件设施的实施例，其中，各种特定类型的功用和功能由交易授权系统的特定实施例执行，并且其中，用户以特定方式定义和使用特定类型的指令规则集，比如授权各种类型的基于费用的程序性交易的付款指令规则集。然而，本领域的技术人员会意识到，本发明的技术可被用于包括除 Web 服务之外的服务和选项以及用于除提供金融支付之外的目的的多种其它情况，并且本发明不限于所提供的示例性详述。

图 2 是示出 Web 服务(“WS”)付费处理系统(“PHS”)金融交易授权系统 200 的示例实施例的构成和信息流的网络图。各种用户和系统与示出示例中的 PHS 交互，以接收比如用于发生在用户和/或系统之间的交易的付款处理功能。在示出的实施例中，PHS 包括帐户系统 220 和 Web 服务交易系统 210，如下所述。

具体地说，在示出的实施例中，执行应用开发的用户(未示出)使用计算系统 250 与 PHS 交互 251，以允许代表用户所开发的应用程序的以后的付款处理功能。应用开发方用户的计算系统 250 首先与帐户系统交互，以定义应用开发方的用户帐户 230，比如通过由帐户系统提供的交互 GUI 224，或通过由帐户系统提供的一个或多个 Web 服务 API 222。在示出的实施例中，每一这样的用户帐户可包括一个或



多个使用指令规则集 235，每一使用指令规则集指定关于使用指令规则集可如何及何时被用于付款目的的各种规则。此外，每一使用指令规则集具有相关的唯一参考权标 240，用于以后参考该使用指令规则集。在应用开发方用户利用一个或多个使用指令规则集创建帐户之后，用户将使用指令规则集的一个或多个参考权标 257 包含在其创建的一个或多个应用程序 255 内。

类似地，向其它提供 Web 服务的一个或多个用户（未示出）使用计算系统 260 通过帐户系统的交互 GUI 和/或 Web 服务 API 与帐户系统交互 261。与应用开发方一样，每一 Web 服务提供方用户可利用帐户系统的一个或多个使用指令规则集创建帐户。然后，Web 服务提供方用户可将其使用指令规则集的一个或多个其参考权标 267 相关为一个或多个 Web 服务提供方交易系统 265 的一部分，利用其 Web 服务提供方向其它提供 Web 服务（例如费用）。

当应用程序 255 之一以后被最终用户计算系统 270 的最终用户（未示出）执行时，应用程序调用 290 由 Web 服务提供方的交易系统提供的一个或多个 Web 服务，比如通过 Web 服务 API 269。然而，在示出的实施例中，仅当对应用程序提供适当的付款时，Web 服务提供方才提供请求的 Web 服务。因此，在示出的实施例中，应用程序提供应用开发方用户的参考权标 257 作为 Web 服务调用的一部分。在将请求的 Web 服务提供给应用程序之前，Web 服务提供方交易系统与 PHS 交易系统 210 联系 295，以获取适当的授权和付款，比如通过调用 PHS 交易系统的 Web 服务 API 212。在示出的实施例中，Web 服务提供方交易系统提供 Web 服务提供方的参考权标 267 以及应用程序的参考权标 257，作为与 PHS 交易系统交互的一部分，并且还提供关于要发生的交易和相关的支付的信息。虽然这里未说明，但在一些实施例中，由 Web 服务提供方的交易系统 265 和参考权标 267 提供的一个或多个 Web 服务可为 Web 服务提供方的一个或多个应用程序的一部分，比如提供的应用程序作为 Web 服务提供方的 Web 站点（未示出）的一部分。

在示出的实施例中，通过首先与帐户系统 220 交互以检索关于对应于 Web 服务 API 212 的调用中提供的参考权标的使用指令规则集 235，以及进一步检索这些使用指令规则集所属的用户帐户 230 的其它信息，PHS 交易系统 210 确定付款交易是否被准许。在获得使用指令规则集和其它信息之后，PHS 交易系统 210 确定规则集是否相容，并且否则授权进行请求的付款，如果是这样，则 PHS 交易系统执行付款（例如，通过对与应用程序的参考权标 257 相关的支付工具收费，以及通过将部分该收费存入在与交易系统 265 的参考权标 267 相关的收支库中），并且向交易系统 265 提供确认，而交易系统 265 随后将最初请求的 Web 服务提供给应用程序。在一些实施例中，PHS 还可获取其提供的一些或所有服务的付款，比如执行的每一交易的部分付款和/或与帐户创建和/或修改相关的付款。

尽管在示出的实施例中 PHS 交易系统 210 和帐户系统 220 被示出为分开的但是合作的系统，但在其它实施例中，也可以其它方式提供示出的功能。例如，在一些实施例中，单一系统可提供帐户创建和维护功用以及交易授权和处理功用，而在其它实施例中，交易系统 210 和帐户系统 220 可由不相关的系统/机构来提供。

此外，尽管这里仅说明了付款型交易，但在其它实施例中，各种其它类型的交易也可被执行，不管是代替付款交易还是另外的付款交易。而且，尽管示出的实施例描述了使用交易系统 265 的 Web 服务的每次使用的费用，但在其它实施例中，可执行各种其它类型的购买交易，比如正在进行的订购（subscription）一个或多个应用程序 255 的交易系统 265 的一个或多个 Web 服务。或者，在将应用程序 255 分发给最终用户之前，比如应用开发方用户可购买 Web 服务的指定次数的使用或指定的时间段期间不限次数的使用，并与参考权标 257 相关。在这种情况下，关于订购或其它的多次使用购买的信息可由帐户系统以各种方式存储，使得交易系统 210 以后能确定付款交易是否被授权。关于利用使用指令规则集自动授权程序性交易的附加细节被包含在共时提交并且名称为“程序性交易的自动授权”的相关美国专利申请

No. \_\_ (律师号 No. 120137.489) 中, 其整个内容以参照的方式被包含在这里。

图 9A-9B 和 10A-10C 均示出执行由第三方交易授权系统自动授权的程序性交易的附加示例(称作用于这些示例的“Subway”)。特别是, 图 9A 和 9B 示出进行基于 Web 服务的程序性交易的机对机微支付的示例。在该示例中, 称作“MapX”的 Web 服务提供方提供 Web 服务, 该 Web 服务用于对每一调用或呼叫以一分的费用提供按地址的地图服务。作为提供给各最终用户消费方的蜂窝电话(或“移动电话”)服务的一部分, “CellX”为该 Web 服务的潜在 Web 服务消费方。在 CellX 签约 MapX 的程序性地图服务之后, CellX 使用该 MapX 服务以使地图可用于其蜂窝电话服务消费方。

图 9A 和图 9B 的示例的示出的一系列活动始于 MapX 的代理(未示出)利用 Subway 交易授权系统 920 执行交互帐户设置活动 931, 特别是利用 Subway 系统 920 提供的 Web 站点 922 以指定关于 MapX 的信息(例如, MapX 的凭证信息和自动交易所(或“ACH”)银行信息)。然后, 通过提供接收的 MapX 的信息 Subway Web 站点 922 与 Subway Web 服务 924 交互, 以创建 MapX 的帐户, 并通过接收新帐户的 ID 作为响应。然后, 通过提供 MapX 的凭证、帐户 ID 和付款指令规则集的规则 933a, MapX 的代理启动与 Subway Web 服务 924 的程序性交互 933, 以创建 MapX 的付款指令规则集, MapX 代理的这种程序性交互可包括使用以后将向消费方提供地图 Web 服务的 MapX Web 服务提供方应用 910, 或者使用另一程序(未示出)。Subway 系统 920 将付款指令规则集存储 926a 在存储器 926 中, 并且向 MapX 代理返回相关的权标, MapX 代理包含 MapX Web 服务提供方应用 910 权标, 以以后用于授权交易。在注册付款指令规则集之后, MapX 现在可以使用 MapX 应用 910 将其地图 Web 服务提供给 Web 服务的潜在消费方。

然后, CellX 的代理(未示出)与 MapX 应用 910 交互 935, 以签约 MapX 地图服务(例如通过使用 MapX 应用 910 的适当的 Web 服

务来执行签约活动),并且作为响应接收 MapX 的权标。通过与 Subway Web 站点 922 交互以指定关于 CellX 的信息,CellX 代理还利用 Subway 系统 920 执行帐户设置活动 937,并且作为响应接收帐户 ID。为了获取帐户 ID,通过提供从 CellX 接收的信息,Subway Web 站点 922 与 Subway Web 服务 924 交互,以创建 CellX 帐户。然后,通过提供 CellX 的凭证、帐户 ID 和付款指令规则集的规则 939a,CellX 代理利用启动与 Subway Web 服务 924 的程序性交互 939,以创建 CellX 的付款指令规则集。然后,Subway 系统 920 将付款指令规则集存储 926a 在存储器 926 中,并且向 CellX 代理返回相关的权标,CellX 代理包括可通过 CellX 的蜂窝电话服务可用于 CellX 消费方的 CellX 应用 905 权标。在注册付款指令规则集之后,CellX 现在可通过 CellX 应用 905 将 MapX 的地图 Web 服务提供给她消费方。

然后,CellX 蜂窝电话服务的消费方在 CellX 应用 905 中选择 941 使用 MapX 应用 910 的 MapX 的地图功能的功能,比如通过选择作为 CellX 应用 905 的在线目录服务的一部分的地图相关的链接。然后,用户选择使 CellX 应用 905 从 MapX 应用 910 程序性调用 MapX Web 服务,以便请求对应于消费方选择的特定地图,而调用包括提供先前与 CellX 应用 905 相关的 CellX 参考权标。然后,MapX 应用 910 向 Subway Web 服务 924 提交包括关于交易、MapX 的凭证、MapX 和 CellX 参考权标的信息的支付授权请求。然后,Subway 系统 920 检索 926b 该参考权标的付款指令规则集,并利用它们确定交易是否被授权,对于该调用执行从 CellX 到 MapX 的付款,并且向 MapX 应用 910 返回指示交易授权的交易 ID。然后,MapX 应用 910 将请求的地图提供给 CellX 应用 905,其将该地图显示给消费方。作为可以实现 MapX 应用 910 和 CellX 应用 905 的一个示例,CellX 应用 905 可以以类似于先前在图 2 中所述的应用程序 255 的方式工作,并且 MapX 应用 910 可以以类似于如先前在图 2 中所述的包括 WS API 269 的 WS 提供方交易系统 265 的应用程序的方式工作。

图 10A-10C 示出使用付款交易规则集授权程序性交易的另一示

例，在该示例中包括用户（名为“Rob”），其通过 Web 服务提供应用 960 以实现第三方商家（merchant）的购物车功能。通过在与 Rob 的购物车应用 960（其保持每一商家的每一消费方的会话信息）交互的商家应用中包含消费方可选的控制，这样的商家可签约并使用 Rob 的购物车功能。当消费方通过商家的应用结帐时，Rob 的购物车应用 960 与 Subway 交易授权系统 970 交互，以处理从消费方到商家的付款。

图 10A 至 10C 的示例的示出系列活动始于商家的代理（未示出），其执行交互活动 981，以通过与 Rob 的购物车应用 960 交互，签约 Rob 的购物车服务，其首先将代理重定向到 Subway 交易授权系统 970，以注册商家的付款指令规则集。下面，商家代理与 Subway 系统 970 提供的 Web 站点 972 交互以签约（在该示例中，商家已有 Subway 系统 970 的帐户），并且将付款指令规则信息 981a 提供到 Subway 系统 970。然后，通过提供 Rob 的购物车 Web 站点应用 960 的凭证、商家的帐户 ID 和付款指令规则集的规则 981a，Subway Web 站点 972 与 Subway Web 站点 974 交互，以创建付款指令规则集。下面，Subway 系统 970 将付款指令规则集存储 976a 在存储器 976 中，并且通过 Rob 的购物车 Web 站点应用 960 向商家返回相关的权标。然后，商家包括与购物车功能相关的消费方可选控制作为商家的 Web 站点应用 955 的一部分，而该控制与适当的 URL 相关，在调用由 Rob 的购物车 Web 站点应用 960 提供的相应的 Web 服务时该 URL 将商家参考权标用作参数。

然后，商家的消费方与商家的 Web 站点应用 955 交互，以选择适当的控制来向消费方的购物车增加项目，这使得信息能够从商家应用 955 被发送到包括消费方的会话 ID 和商家的参考权标的 Rob 的购物车 Web 站点应用 960。当消费方以后通过商家 Web 站点应用 955 指示进行结帐时，商家应用 955 将信息发送给 Rob 的购物车应用 960，其与 Subway Web 站点 972 交互，以便针对对应于购买消费方购物车中项目的量动态地指定消费方的付款指令规则的一次的集合。此外，提示消费方在 Subway Web 站点 972 签约，包括创建帐户（如果需要的话）。然后，通过提供 Rob 的购物车 Web 站点应用 960 的凭证、消费方的

帐户 ID 和付款指令规则集的规则 983a, Subway Web 站点 972 进行与 Subway Web 服务 974 的交互, 以创建付款指令规则集。下面, Subway 系统 970 将付款指令规则集存储 976a 在存储器 976 中, 并将消费方的相关权标返回给 Rob 的购物车 Web 站点应用 960。

当商家的消费方通过商家 Web 站点应用 955 指示发出选择“确认命令”控制的命令时, 商家应用 955 将包括会话 ID 的指示发送给 Rob 的购物车 Web 站点应用 960。然后, Rob 的购物车 Web 站点应用 960 检索先前被创建的消费方的参考权标, 并向 Subway Web 服务 974 提交包括关于交易、购物车 Web 站点凭证、和商家及消费方的参考权标的信息的支付授权请求。下面, Subway 系统 970 针对参考权标检索 976b 付款指令规则集, 并使用它们来确定交易被授权, 执行交易的从消费方到商家的付款, 并向 Rob 的购物车 Web 站点应用 960 返回指示交易授权的交易 ID。

图 1A 至 1B 提供了通过易授权和处理系统创建以后使用帐户和使用指令规则集的用户交互的示例。特别是, 图 1A 提供 GUI 的示例, 通过 GUI, Web 服务提供方用户 ABC 交互地创建用户帐户, 比如通过在由帐户系统提供的 Web 页中填入表格。在该示例中, 示例交互创建屏幕包括具有概述信息的标题区 111, 下面为部分 113, 在部分 113 中, 用户可指定帐户的各种一般信息, 比如帐户名、用于访问控制以观看并修改帐户的密码、任何可选的认证和任何可选的组织关系 (organization affiliation)。在该示例中, 用户 ABC 指定来自第三方公司 BCD 协会的认证, 因为由 ABC 提供的 Web 服务的潜在消费方的一些使用指令规则集可能请求这样的认证以便授权向 ABC 付款。类似地, 用户 ABC 指示与其自己公司和各种 Web 服务提供方联盟的组织关系, 因为 ABC 的 Web 服务的潜在消费方的使用指令规则集可能类似地请求一个或多个这样的组织关系以授权向用户 ABC 付款。

在该示例中, 帐户创建屏幕还包括可选部分 114, 在可选部分 114 中, 用户 ABC 可指定一个或多个收支库, 用于用户可能收到的比如用于向消费方提供基于费用的 Web 服务的付款。在该示例实施例中, 用

户 ABC 指定两个收支库，其中可存入收入的付款，包括银行帐户和借记卡。用户还指定了用于借记卡的限制，使得存入的任何付款必须为最小值至少 5 分。在示出的实施例中，部分 114 还指示正创建的用户帐户本身可用作用户的收支库。

帐户创建屏幕还包括部分 115，在部分 115 中，用户 ABC 可指定要被用于对其它方进行付款的一个或多个支付工具。在该示例中，用户 ABC 指定同一银行帐户作为可用的支付工具，尽管具有来自帐户的任意单独付款不大于 \$50 元的限制，而且信用卡支付工具具有最大单独付款 \$200 元和最大累积结算 \$2000 元的限制。类似地，部分 115 指示正创建的用户帐户可被用作用户的支付工具，其用于该用户并且在该示例中包含最大 \$1000 元结算信用的限制。指定各种类型的信息之后，用户可随后选择控制 116b 以创建帐户，或者用户可选择控制 116a 以取消帐户创建。

在示出的实施例中，在最初创建用户帐户之后，通过图 1B 中示出的示例屏幕向用户呈现创建一个或多个使用指令规则集的选项。特别是，示出的屏幕包含具有概述信息的标题部分 117 和规则说明部分 118，在规则说明部分 118 中，可针对正创建的使用指令规则集指定各种信息。在该示例中，部分 118 允许用户指定正创建的使用指令规则集的名称、规则集的使用类型（其在该示例中为接收付款）、接收任何这样的付款的收支库和在该示例中未被指定的到期数据。用户 ABC 还可指定该规则集的一个或多个特定规则。该示例示出已被定义的两个规则，以及用于示出其它规则属性的第三规则的下拉菜单，在该示例中，其可针对该使用指令规则集被选择和定制。

例如，用户 ABC 已指定该使用指令规则集将仅接收满足第一“允许的付款人”角色属性的用户的付款，在该示例中，这可根据付款人的指定组织关系、付款人的分级、或指示其 ID 的特定付款人来满足。如所指示的，在该示例中可被指定的其它规则属性包括：用户 ABC 将不接受其付款的被禁止的或阻止的付款人、可使用使用指令规则集的允许的或禁止的交易类型的说明、接收付款的支付工具和货币的允许的

或禁止的类型、用于以后处理发生的任何纠纷的条款和条件等。此外，示出的实施例允许用户指定用户定义的规则，比如通过包含可执行代码和/或用于调用可访问的可执行功能（例如 Web 服务）的机制。在指定正创建的使用指令规则集的各种信息之后，用户 ABC 可选择部分 109 中的控制之一，以创建当前的使用指令规则集并继续创建帐户的另一规则集、创建当前的使用指令规则集并退出帐户修改、或跳过创建当前指令规则集并退出。

出于示例目的，以简化方式设置该示例所示的规则属性的类型，在其它情况下，可以以其它方式指定类似信息。例如，不是使允许的和禁止的付款人具有不同的属性，提供付款人的单一属性，使用评估标准的类型来确定用户是允许的还是禁止的（例如，为了满足规则，指定付款人“是”或“不是”指定值）。关于示例属性的附加细节另外被描述。

图 1C 示出创建用户帐户的使用指令规则集的可选技术。特别是，在该示例实施例中，用户帐户和使用指令规则集由 Web 服务消费方应用开发方用户 CDE 程序性地创建，比如通过由 WS 付款处理系统提供的一个或多个 Web 服务。在该示例实施例中，作为程序性调用的一部分提供的信息包括图 1C 中示出的示例 XML 文件 131。该示例 XML 文件指定要与正创建的使用指令规则集相关联的帐户 ID（例如响应创建用户帐户的较早的程序性调用由用户所接收的帐户 ID），并且还提供类似于关于图 1B 以交互方式指定的各种信息。

特别是，该使用指令规则集正被用户 CDE 创建，用于提供交易的付款，该使用指令规则集将通过由用户 CDE 创建的一个或多个应用程序比如基于费用的 Web 服务（例如，由 Web 服务提供方用户 ABC 提供的一个或多个 Web 服务）被调用。在该示例中，正创建的使用指令规则集将使用用户 CDE 的 PHS 帐户（在该示例中命名为“我的 PHS 帐户”，并具有指定的到期日期。针对使用指令规则集定义的两个规则的第二规则还利用适当的评估类型和规则值限定总付款，通过使用预定规则属性“允许的付款”。该使用指令规则集可授权小于或等于\$1000



元的量。为使用指令规则集指定的第一规则识别特定付款接受方，对其不能使用使用指令规则集，在该示例中，其被指定为一般属性“接受方”和适当的评估类型“IS-NOT”，而不是更特定属性类型“不允许的接受方”。在其它实施例中，可以以各种其它方式，包括以使用其它格式的 XML 文件和/或以除 XML 以外的方式，提供相应的信息。

图 1C 也示出了用于创建用户 CDE 的帐户的另一使用指令规则集的第二 XML 文件 133，以便伴随 WS 付款处理系统的第二程序性调用。特别是，该第二使用指令规则集示出用于除付款处理之外类型的交易的使用指令规则集的示例，以便在该示例中指定访问控制信息。例如，用户 CDE 可具有提供对一个或多个程序性服务或信息源进行访问的签约或其它会员资格，但用户 CDE 可能希望限制何时这样的访问权可用于用户所提供的一个或多个应用。在该示例中，正在 XML 文件 133 中创建的使用指令规则集不包括相关的支付工具，但包括各种访问信息，其包括要被提供的用户名和密码以按照签约/会员资格获得访问，在其它实施例中以及在其它情况下，可指定和使用各种其它类型的访问控制信息。

该使用指令规则集还示出定义用户定义规则的示例，在该示例中，对其给定用户指定的名称“允许的使用”（例如，以后用于独立于任何其它用户定义规则引用该用户定义规则，未示出），以及基于 URI 的调用说明，当其如指示的被执行时，其将提供响应指示规则是否被满足。在该示例中，调用说明包括以与对特定的潜在交易特定的信息要被填充的参数的指示，所述填充可这样进行，将置于方括号内的参数“access-provider.name”替换成针对访问提供方（对特定的交易该指令规则集与其匹配）的使用指令规则集的帐户的对应名称。这样的用户定义规则可进行关于在各种方式下的特定情况下规则是否被满足，以及是否使用包括关于使用使用指令规则集的应用和/或执行应用程序的最终用户的各种类型的信息的确定。类似地，用户定义规则可在其它实施例中以各种其它方式被定义和调用，包括通过提供作为创建使用指令规则集的一部分的可执行代码（例如脚本或小应用程序），而

其将与用户定义规则相关联。

图 1D 和 1E 示出几个用于存储关于已创建的用户帐户和使用指令规则集的信息的示例数据库。特别是,图 1D 示出示例帐户数据库 140,其存储关于已创建的用户帐户的各种信息,而每一项 140a 至 140b 对应于不同的用户帐户。在该示例中,每一项不仅包括由创建帐户的用户指定的信息,而且包括针对帐户自动生成并与帐户相关的信息。例如,项 140a 对应于先前在图 1A 已描述其创建的用户帐户,而列 143 至 146 包括由图 1A 的部分 113 中的用户指定的信息。此外,当帐户被创建时,PHS 生成帐户的唯一帐户 ID,其被存储在列 141 中。此外,在该示例中,用户 ABC 在创建图 1A 中的 PHS 帐户之前已被 PHS 或相关的系统所知,并且列 142 包括事先与用户相关的用户 ABC 的用户 ID (例如作为事先为用户 ABC (比如 Web 商家或商店)创建的用户帐户的一部分)。

此外,如先前就图 1A 所述的,在该示例实施例中,PHS 向至少一些用户提供将他们的 PHS 帐户用作收支库和/或支付工具的能力。因此,列 149 和 150 包括对应于用户的 PHS 帐户的该方面,包括帐户的当前结余和帐户所允许的最大结余(例如在帐户创建时以用户特定方式动态确定以及根据用户后来的经验适当地可调整的最大结余)的信息,----然而,由于在该示例中用户 ABC 不将 PHS 帐户用作收支库或支付工具,因此当前帐户结余为零。虽然这里未示出,但各种其它类型的信息也可与帐户相关,比如先前已对帐户执行的交易的指示、帐户已出现的变化、其它用户和对那些用户访问帐户的相关访问控制的指示等。项 140b 包含先前就图 1C 所述的用户 CDE 的帐户的类似对应信息。

帐户数据库 140 的列 147 和 148 包括关于为了用于用户帐户而定义的任何收支库和支付工具的信息。具体地说,项 140a 的列 147 中的信息对应于在图 1A 的部分 114 中指定的收支库,项 140a 的列 148 中的支付工具信息类似地对应于在图 1A 的部分 115 中指定的支付工具。在该示例中,每一收支库和支付工具被给定唯一的 ID,该 ID 被存储

在列 147 和 148 中，而附加的示例数据库收支库数据库 180 和支付工具数据库 190 分别存储关于用户指定的收支库和用户指定的支付工具的信息。在该示例实施例中，收支库数据库和支付工具数据库均包含分别对应于单独的收支库和支付工具的多个项 180a 至 180b 和 190a 至 190c。

此外，数据库 180 和 190 存储每一库或工具的各种信息，包括列 181 和 191 中的自动生成的 ID、列 182 和 192 中的与收支库或支付工具相关的帐户的指示、列 183 和 193 中的库或工具的用户指定名称、列 184-186 和 194-196 中的关于如何将收支库或支付工具分别用于支付存入或付款的各种信息。在其它实施例中，可提供并使用各种其它类型的附加信息，比如以对工具和/或库特定类型的方式（例如，使用与信用卡相关的用户的全名以便进行收费，而不将该信息用于银行帐户）。此外，在示出的示例中，用户可定义关于使用收支库或支付工具的一个或多个限制，如先前所述，而在该示例中这样的限制信息被存储在列 187 - 188 和 197 - 198 中。

虽然用户 ABC 使用单一银行帐户作为收支库和支付工具（分别对应于项 180a 和 190a），但在该示例中，对于这些使用的每一个，银行帐户被给定不同的 ID（如这些项的列 181 和 191 中所示），尽管在其它实施例中可代替使用单一 ID。此外，虽然这里未示出，但在其它情况下，单一付款源或目的地（例如银行帐户）可与多个定义的收支库和/或多个定义的支付工具一起使用，比如用来在该付款源或目的地的不同使用的情况下定义不同的限制或其它条件。此外，在该示例中，用户 CDE 还未被指定任何用户指定的收支库或用户指定的支付工具，而是将 PHS 帐户用作支付工具（因为用户 CDE 当前未收到任何付款，所以他/她当前不需要收支库）。

图 1E 示出存储关于各种定义的使用指令规则集的信息的使用指令规则集数据库 160。具体地说，项 160a 对应于图 1B 中由用户 ABC 定义的使用指令规则集，项 160b 对应于图 1C 中由 XML 文件 131 定义的使用指令规则集。在该示例中，每一使用指令规则集被给定唯一

ID (如列 161 中所示), 并且还被提供自动生成的唯一参考权标, 用于以后参考使用指令规则集 (如列 163 中所示)。尽管在该示例中参考权标为长随机数, 但在其它实施例中, 各种其它类型的信息可被用于参考权标, 通常可包括能参考或指示付款指令规则集的任何信息。在一些实施例中可被使用的参考权标的其它类型的示例包括数字证书或其它凭证、密钥对的公钥或私钥 (如果 PHS 比如通过与用户的帐户相关访问其它密钥) 和/或使用这些密钥之一加密的信息 (例如, 与参考权标相关的使用指令规则集的拷贝、关于参考权标所属用户的信息、关于正对其使用参考权标的潜在交易的信息等)。数据库 160 还包括使用指令规则集的各种其它指定信息, 包括使用指令规则集与其相关的帐户的帐户 ID (如列 162 中所示)、名称和使用类型 (如列 164 和 165 中所示)、可选截止日期 (如列 167 中所示) 和与使用指令规则集一起使用的一个或多个可选收支库或支付工具的指示 (如列 166 中所示)。

数据库 160 还包括列 168 中的使用指令规则集的规则的指示, 在该示例中, 其用对应于这些规则的规则 ID 被示出。然后, 关于规则的附加信息被提供在使用指令规则集数据库 170 中。具体地说, 在该示例中每一规则均具有相关的唯一规则 ID (如列 171 中所示) 和与规则相关的使用指令规则集的 ID 的指示 (如列 172 中所示)。列 173 至 177 包括关于如何确定在特定情况下规则是否被满足的各种信息, 并且在其它实施例中和在这种规则的其它情况下, 可类似地存储各种其它类型的信息, 比如以对不同类型的规则特定的方式 (例如通过存储用户定义的规则的调用细节)。

图 3 示出适于执行交易授权处理系统设施 340 的实施例的服务器计算系统 300, 以及分别针对 Web 服务消费方和提供方的计算系统 350 和 370。服务器计算系统包括 CPU 305、各种 I/O 设备 310、存储器 320 和内存 330。I/O 设备包括显示器 311、网络连接 312、计算机可读介质驱动器 313 和其它 I/O 设备 315。

交易授权处理系统的实施例在内存里执行, 并且其包括交易确认

器组件 341、交易处理器组件 343、帐户管理器组件 345、可选付款集中器组件 347 和可选安全管理器组件 349。具体地说，交易授权处理系统接收潜在交易的指示并确定是否授权交易。这样的潜在交易可包括在 Web 服务消费方系统 350 的内存 357 中执行的应用程序 359 和在 Web 服务提供方系统 370 的内存 377 中执行的 Web 服务服务器 379 之间的交易，和/或在一个或多个这样的系统 350 和 370 与一个或多个其它计算系统 390 之间的交易。

在执行要被授权的潜在交易之前，用户（例如计算系统 350 和 370 的用户）与帐户管理器组件交互以创建用户帐户，而关于用户帐户的信息 325 被存储在存储器 320 中。每一这样的用户也可与帐户管理器交互以创建一个或多个使用指令规则集，而关于每一这样的指令规则集的信息 327 以与用户的帐户相关的方式被存储。帐户管理器还创建所创建的每一这样的使用指令规则集的参考权标，而关于参考权标的信息 328 以与使用指令规则集相关的方式被存储。在创建的参考权标被提供给用户比如计算系统 350 和 370 的用户之后，这些用户可将参考权标 356 和 376 分别存储在计算系统的存储器 351 和 371 中，以以后分别用于应用程序 359 和 Web 服务服务器 379。尽管在示出的实施例中帐户管理器组件是交易授权处理系统的一部分，但在其它实施例中，一些或所有帐户管理器组件功能可由与交易授权处理系统交互的另一系统代替执行。

当交易授权管理机系统 340 收到授权确定的潜在交易和交易各方的两个或多个参考权标的指示时，交易确认器组件检索关于对应于该参考权标的使用指令规则集的存储信息，并确定这些使用指令规则集在当前条件下是否满足潜在交易。如果是的话，交易处理器组件参与执行或处理该潜在交易，以便安排基于费用的交易的各方之间的金融支付执行。

在一些实施例中，交易授权处理系统还可包括可选付款集中器组件 347 和可选安全管理器组件 349 的一个或两者。付款集中器组件（当存在时）可集中正进行对适当各方的付款和/或从适当各方接收的付

款,比如基于这些各方的偏好和/或使用其它信息。安全管理器组件(当存在时)可执行各种类型的附加的安全相关的验证,以便确认交易各方的身份,确认提供参考权标的用户的身份和/或授权,执行对试图创建和/或修改用户帐户及使用指令规则集的用户的用户授权等。

在不同实施例中可以以各种方式访问交易授权处理系统的功能。例如,一些用户可物理访问服务器计算系统,如果这样的话,用户可与各种 I/O 设备 310 交互以提供和/或接收信息。或者,其它用户可使用客户计算机系统以直接访问系统,比如远程地通过网络 380(例如通过因特网和/或万维网)。这种用户可使用软件或提供在客户计算机系统上的其它功能(比如在内存 360 中执行的浏览器(未示出))来与交易授权处理系统交互。在其它实施例中,用户可通过与一个或多个其它计算机的交互间接地从交易授权处理系统接收功能和/或信息,所述一个或多个计算机可直接接收来自交易授权处理系统的功能或信息并将其提供给用户。此外,一些或所有交易授权处理系统组件可向用户提供各种反馈或其它一般类型的信息(例如响应用户请求),并且该信息可在服务器系统的显示器 311 上或在客户机系统的 I/O 设备 354 或 374 之一上呈现给用户。

本领域的技术人员会意识到,计算系统 300、350 和 370 仅是示例性的,不是意在限制本发明的范围。计算系统 300 可被连接到未示出的其它设备,包括通过一个或多个网络比如因特网或通过万维网(“Web”)。更一般地,“客户机”或“服务器”设备可包含可交互的硬件或软件的任意组合,包括计算机、网络设备、因特网装置、PDA、无线电话、蜂窝电话、电子设备(organizer)、基于电视的系统和包括互通能力的各种其它消费产品。例如,计算系统 350 之一可为蜂窝电话、无线 PDA 或其它便携设备,并且该电话/PDA/设备的用户可进行导致使用交易处理系统 340 的功用的一个或多个选择,以便利用一个或多个提供方计算系统 370 处理交易。此外,在一些实施例中,由示出的交易授权处理系统组件提供的功能可被组合在较少的组件中或分布在附加的组件中。类似地,在一些实施例中,一些示出的组件的

功能可不被提供，并且/或者其它的附加功能可以是可用的。

本领域的技术人员还会意识到，尽管各种项目被示出为在使用时被存储在内存或存储器中，但出于内存管理和数据完整性的目的，这些项目或其部分可在内存和其它存储设备之间被传送。或者，在其它实施例中，一些或所有软件模块和/或组件可在另一设备的内存中执行，并且通过计算机间的通信与示出的计算设备进行通信。一些或所有交易授权处理系统组件或数据结构也可被存储（例如作为指令或结构数据）在计算机可读介质，比如硬盘、内存、网络或要由适当的驱动器读取的便携产品中。交易授权处理系统组件和数据结构也可作为一般数据信号（例如作为载波的一部分）被发送到各种计算机可读传输介质，包括基于无线和基于线/缆的介质上。因此，可利用其它计算机系统配置实现本发明。

图 4 是用户帐户管理器例程 400 的实施例的流程图。该例程允许用户创建和修改用于交易授权和处理系统的帐户，包括定义用于帐户的使用的使用指令规则集。在一些实施例中，例程可作为用户可与其交互的交互用户接口的一部分（例如作为 Web 站点的一个或多个网页的一部分）被实现，而在其它实施例中，例程可响应以用户的名义一个或多个程序性调用（例如作为以用户的名义访问的一个或多个 Web 服务）被执行。

例程从步骤 405 开始，其中接收与帐户有关的指令。在步骤 410 中，例程确定指令是否与创建帐户有关，如果是，则继续进到步骤 415 以接收与定义帐户有关的各种信息。例如，在一些实施例中，用户可提供帐户名、帐户密码或其它安全访问机制、用于帐户的一个或多个支付工具的指示、用于帐户的一个或多个收支库的指示等。然后，例程继续进到步骤 420 以自动生成与帐户的使用有关的附加信息，比如在一些实施例中的唯一帐户 ID、如果帐户可被用作可包含付费的支付工具的话对用户允许的最大结余的确定等。在步骤 425 中，例程利用接收的和生成的帐户信息创建帐户，并且在步骤 430 中将新帐户的参考提供给帐户的创建方（例如帐户 ID 的指示）。

如果在步骤 410 中确定收到的指令不与创建帐户有关，则例程继续进到步骤 435 以接收与修改帐户有关的信息，并且在步骤 440 中如指示的适当地修改帐户。在步骤 445 中，例程确定帐户修改是否包括创建或修改使用指令规则集。如果是，则例程继续步骤 450 以生成与使用指令规则集相关的唯一参考权标，以允许以后参考指令规则集。在步骤 455，例程向帐户所有者提供参考权标的指示，用于以后参考使用指令规则集。在步骤 430 或 455 之后，或者如果在步骤 445 中确定帐户指令不与创建或使用指令规则集有关，则例程继续进到步骤 460 以确定是否进行当前帐户的修改。如果是，则例程返回到步骤 435，如果否，则例程继续进到步骤 499 并结束。

尽管这里未示出，但在其它实施例中可提供与用户帐户有关的附加功能，以便允许用户通过提供关于帐户的信息审阅帐户。此外，在一些实施例中，可使用附加的安全控制以确保试图创建和/或修改帐户的用户被授权这样做。此外，在一些实施例中，用户帐户可与公司或不同于提供交易授权和处理功能的系统的实体相关。

图 5 是交易授权和处理例程 500 的实施例的流程图。例程接收来自多个交易方的交易被授权或被处理的指示以及对应于使用指令规则集的参考权标的指示，并且当使用指令规则集相容时确定授权交易。在一些实施例中，例程被程序性调用，比如通过一个或多个 Web 服务，尽管在其它实施例中可以以其它方式调用例程（例如交互地通过图形用户接口）。

例程从步骤 505 开始，其中接收多方交易和多方的每一个的参考权标（例如交易双方或多方的每一个的参考权标）的指示。然后，例程继续进到步骤 510 以检索对应于权标的使用指令规则集，并且在一些实施例中还可从一些或所有这些使用指令规则集所属的用户帐户检索其它信息或使其它信息可用。

然后，例程继续进到步骤 515，以确定是否执行安全验证，从而确定是否允许各方使用该权标，比如根据由各方提供的附加安全信息。如果是，则例程继续进到步骤 520 以尝试获取并验证表示授权各方使



用权标的安全信息，并且在步骤 525 中，确定这样的授权是否被验证。在其它实施例中，比如当使用的权标本身的指令规则集包含控制使用指令规则集何时被允许使用的规则或其它限制时，使用这种权标的这样的安全性或其它访问授权不被执行。

如果在步骤 525 中确定权标使用授权被验证，或者如果在步骤 515 中确定未执行安全验证，则例程继续执行子例程 530 以基于检索的使用指令规则集确认 (validate) 交易，如就图 6 更详细所述。如果在步骤 535 中确定交易被确认，则例程继续执行子例程 545 以尝试执行指示的交易（例如进行基于费用的交易的金融支付），比如对交易类型特定的方式，如就图 7A 和 7B 更详细所述。在其它实施例中，例程可仅确定交易是否被授权，而交易的执行留给其它方。

在步骤 545 之后，如果在步骤 550 中确定指示的交易执行成功，则例程继续进到步骤 555 以返回成功的指示，比如在例程 500 被调用作为 Web 服务的一部分的情况下以响应 XML 文件。如果在步骤 525 中确定权标使用授权未被验证，或者在步骤 535 中确定交易未被确认，或者在步骤 550 中确定交易执行未成功，则例程继续进到步骤 540 以返回错误消息，比如在例程被调用作为 Web 服务的一部分的情况下作为响应发送的 XML 文件。在步骤 540 或 555 之后，例程继续进到步骤 599 并结束。

图 6 是交易确认子例程 530 的实施例的流程图。该子例程接收交易和由交易各方指定的两个或多个使用指令规则集的指示，并确定是否根据使用指令规则集的规则和其它信息确认交易。

子例程从步骤 605 开始，其中接收交易和两个或多个使用指令规则集的指示。在步骤 610 中，指定与使用指令规则集可使用的交易有关的标准和/或指定关于使用指令规则集可使用的交易的各方的信息的每一规则被选择，并且关于交易和/或其它各方的信息被检索，以便确定规则是否被满足。例如，为了获得关于交易的另一方的信息，可获取来自与其它方的帐户相关的用户帐户的信息（例如与来自其它方的权标的使用指令规则集相关的用户帐户），以及关于可以其它方式

访问的其它各方的信息（例如基于交易处理例程的调用请求，比如根据用作 Web 服务调用的一部分的 HTTP 标题）。类似地，基于作为例程调用的一部分（例如作为提供给交易的交易对象或文件的一部分）接收的信息或来自外部源的信息的关于交易的信息可以是可用的。

在步骤 610 之后，子例程继续进到步骤 615，以识别不是用户定义的规则以及指定与可允许交易有关的其它条款或条件的其它指定规则，并且确定任意这种指定规则是否与来自其它指定的使用指令规则集和/或交易本身相适合。然后，子例程继续进到步骤 620，并且以对这些规则指定的方式调用任意用户定义的规则，并获取指示这些规则是否被满足的响应。在步骤 620 之后，例程继续进到步骤 630 以确定是否指定的使用指令规则集的所有规则被满足。如果是，则子例程继续进到步骤 640 并返回交易被确认的指示，如果否，则子例程继续进到步骤 635 以返回交易未被确认的指示。在步骤 635 或 640 之后，子例程继续进到步骤 699 并结束。

在其它实施例中，交易确认可以以其它方式来执行，比如通过以不同命令评估不同类型的规则，和/或在发现任何规则不被满足的情况下停止进一步的规则评估。或者，在其它实施例中，可以以除要求一致满足所有指定规则之外的方式执行交易是否被确认的确定，比如在使用指令规则集使其本身在这些规则集的规则中和/或基于其它的预定标准（例如必须满足的指定规则的百分率或子集的最小阈值）指定其它类型的相互关系的情况下。

图 7A 和 7B 是执行付款处理交易子例程 545 的实施例的流程图。该例程为可对一些类型的交易执行的一种类型的处理的示例，在本例中为付款处理交易，其中交易的至少一部分为向交易的至少一个其它方提供付款。

子例程从步骤 705 开始，其中接收付款交易（例如付款量和可能的与付款有关的条款）和至少一个付款提供方和至少一个付款接受方的每一个的参考权标的指示。在步骤 710 中，子例程对每一付款提供方确定要被使用的支付工具和该支付工具的任何定义的付款限制，比

如基于与对应于由该付款提供方提供的权标的使用指令规则集相关的支付工具。类似地，在步骤 715 中，子例程确定要被用于每一付款接受方的收支库以及任何定义的付款限制。在步骤 720 中，子例程确定交易指示的一个或多个付款是否满足支付工具和/或收支库的任何定义的付款限制，比如基于最小或最大的当前费用、最小或最大的累积结余等。

在步骤 720 之后，子例程继续进到步骤 725 以确定付款是否满足任何定义的付款限制，如果是，则例程继续进到步骤 735，以针对每一这样的付款确定是分别对支付工具收取费用还是与其它付款一起集中支付（比如基于定期（例如每月）），或者直到达到最小累积收费为止。如果确定不对付款集中收费，则子例程继续进到步骤 740，以立即对付款的对应的支付工具收费，而如果在步骤 735 中确定集中收费，则子例程继续进到步骤 745，以存储用于以后从该支付工具集中收取费用的收费的指示。尽管未示出，但如果出现多个付款，则可对交易的每一支付多次重复步骤 735 至 745。

在步骤 740 或 745 之后，子例程继续进到步骤 750，以确定是将每一付款直接存进对应的收支库中，还是在存入集中付款之前将付款连同其它付款一道集中。如果确定不集中付款，则子例程继续进到步骤 755，以将每一这样的付款立即存到适当的收支库中，而如果在步骤 750 中确定集中付款，则子例程继续进到步骤 760，以存储用于以后集中存入适当的收支库中的付款的指示。尽管这里未示出，但对交易的要进行的每一付款均可执行步骤 750 至 760。

在步骤 755 或 760 之后，子例程继续进到步骤 765 以返回付款成功的指示。如果在步骤 725 中确定交易所指示的付款不满足一个或多个定义的付款限制，则子例程继续进到步骤 730 以返回付款不成功的指示。在一些实施例中，当确定任何定义的付款限制是否被满足时，考虑集中付款和/或存款的能力，以便使多个付款或存款能够满足最小量的限制，而在其它实施例中，单独考虑每一这样的付款。或者，在其它实施例中，可不执行付款存入集中和/或付款收费集中。在步骤 730

或 765 之后，子例程继续进到步骤 799 并结束。

本领域的技术人员还将意识到，在一些实施例中，可以以可选方式提供上述例程提供的功能，比如分割成更多的例程或合并成较少的例程。类似地，在一些实施例中，比如当其它示出的例程分别缺少或包括这样的功能或当所提供的功能数量被改变时，示出的例程可提供比所描述功能更多或更少的功能。此外，虽然可以以特定方式（例如以串行或并行方式）和/或以特定顺序说明执行的各种操作，但本领域的技术人员会意识到，在其它实施例中，可以以其它顺序并以其它方式执行操作。本领域的技术人员还会意识到，上述的数据结构可以以不同的方式构成，比如将单数据结构分割成多数据结构或者将多数据结构合并成单数据结构。类似地，在一些实施例中，比如当其它示出的数据结构分别缺少或包括这样的信息或当所存储的信息数量或类型被改变时，示出的数据结构可存储比所述更多或更少的信息。

下面提供定义和使用使用指令规则集的技术以及指令规则集的规则的例证性示例，尽管本发明并不意在受这些细节所限制。

限定特定付款发送方或接受方的交易的示例如下：

#### 允许的接受方权标

```
recipientToken == 'token1';
```

如果接受方的权标正好与 token1 匹配，则交易将被允许。该规则对所有各方（发送方、接受方、呼叫方）均是可用的。指定多个允许的接受方权标的示例包括如下：

```
RecipientToken in ('token1', 'token[sum]', 'token3');
```

或者,在可选实施例中

```
recipientToken == 'token1' or recipientToken == 'token2' or recipientToken == 'token3';
```

#### 禁止的发送方权标

```
senderToken != 'token1';
```

如果发送方的权标与 token1 匹配，则将不允许交易。该规则对所有各方（发送方、接受方、呼叫方）均是可用的。指定多个禁止的发送方的示例包括如下：

```
sender-Token notIn ( ' token1,' token2',' token3');
```

或者,在可选实施例中

```
senderToken != ' token1' and senderToken != ' token2' and senderToken != ' token3';
```

创建保密握手的示例如下：

该示例是发送方和接受方交换保密握手以商定一些条款。如果双方不具有像这样定义的规则，则将不允许交易。

### 发送方保密

```
string SharedSecretSender := 'rumplestilskin';
```

```
SharedSecretSender == SharedSecretRecipient;
```

对于保密握手规则，如果发送方和接受方未定义同一值，则交易将被禁止。任何字符串可用来代替‘rumplestilskin’，并且任何参考名可用来代替 SharedSecretSender 和 SharedSecretRecipient，只要其不是预定的参考。

### 接受方保密

```
string SharedSecretRecipient := 'rumplestilskin';
```

```
SharedSecretSender == SharedSecretRecipient;
```

对于保密握手规则，如果发送方和接受方未定义同一值，则交易将被禁止。任何字符串可用来代替‘rumplestilskin’，并且任何参考名可用来代替 SharedSecretSender 和 SharedSecretRecipient，只要其不是预定的参考。

创建限定的使用参考权标的示例如下：

付款指令规则集可限定权标可用来进行付款的次数、使用权标的

总量、或权标的截止日期。这些规则仅应用到该权标的使用，因此发送方的付款指令规则集可仅限定使用发送者的权标。类似地，上述规则可应用到呼叫方和接受方。

### 单一使用权标

`MyTokenUsecount <= 1;`

或者,在可选实施例中

`tokenTransactioncount == 1;`

该规则使权标在已使用一次后停止工作。其仅应用到其本身的付款指令规则集，因此发送方的指令集可仅限定该使用。

### 限定使用的次数

`MyTokenliseCount <= 5;`

或者,在可选实施例中

`tokenTransactioncount <= 5;`

该规则使权标在已使用 5 次后停止工作。

### 限定总量

`MyTokenUseTotal Amount <= 'USD 50';`

或者,在可选实施例中

`tokenTransactionAmount <= $50;`

该规则限定该权标被用于 USD 50 的总量。

### 截止日期

`TransactionTimestamp < '30-Jan-2005';`

或者,在可选实施例中

`transactionRequestDate < dateTime '30-Jan-2005';`

该规则使权标在 2005 年一月 30 日停止工作。从该日的午夜 (00:00 GMT) 之后，使用该权标的所有交易被禁止。

### 仅用作接受方

```
MyRole == 'recipient';
```

该规则使权标禁止付款，除非该权标被用作接受方的权标。

### 仅用作呼叫方

```
MyRole == 'caller';
```

该规则使权标禁止付款，除非该权标被用作呼叫方的权标。

### 仅用作发送方

```
MyRole == 'sender';
```

该规则使权标禁止付款，除非该权标被用作发送方的权标。

限定交易量的示例如下：

交易量限定规则可用于所有各方。

### 最大交易量

```
transactionAmount <= 'USD 50';
```

该规则限定交易量为 **USD 50** 或更少。如果交易大于该量，则该规则将禁止交易。

### 最小交易量

```
transactionAmount >= 'USD 1';
```

该规则限定交易量为 **USD 1** 或更多。如果交易少于该量，则该规则将禁止交易。

### 限定交易货币

```
getCurrency ( transactionAmount ) == 'JPY';
```

该规则限定交易货币为日圆。如果交易货币不是日圆，则交易将

被禁止。

付款提供方/发送方指定付款细节的示例如下：

发送方指定付款方法，比如是由信用卡、ACH 银行帐户还是由现有的帐户余额进行付款。

### 余额转帐

```
string paymentMethod := 'balance transfer' ;
```

该规则指定将从现有的帐户余额提取资金。

### 缺省 ACH 银行帐户

```
string paymentMethod := 'ach' ;
```

该规则指定将使用帐户的缺省 ACH 银行帐户。可以以各种方式比如通过 `CrateAccount` 或 `SetDefaultFinancialInstrument` 消息来指定缺省 ACH 帐户。

### 特定 ACH 银行帐户

```
string paymentMethod := 'ach' ;
```

```
string paymentInstrument := 'financialInstrumentReference' ;
```

该规则指定由 `financialInstrumentReference` 所指的特定 ACH 银行帐户。金融凭证参考从 `installACHBankAccount` 消息中被返回。在该示例中，金融凭证参考不是银行帐户号，出于保密原因，其在一些实施例中可不被包括在付款指令规则集中。在其它付款指令规则集中，付款方法应该被指定为允许与其相反的断言。

### 缺省信用卡

```
string paymentMethod := 'credit card' ;
```

该规则指定将使用帐户的缺省信用卡。可以以各种方式比如通过 `CrateAccount` 或 `SetDefaultFinancialInstrument` 消息来指定缺省信用



卡。

## 特定信用卡

```
string paymentMethod := 'credit card';
```

```
string paymentInstrument := 'financialInstrumentReference';
```

该规则指定由 `financialInstrumentReference` 所指的特定信用卡银行帐户。支付工具参考从 `installCreditCard` 消息中被返回。在该示例中，金融凭证参考不是信用卡号，出于保密原因，其在一些实施例中可不被包括在付款指令规则集中。在其它付款指令规则集中，付款方法应该被指定为允许与其相反的断言。

付款接受方指定争端解决策略的示例如下：

当发送方就特定交易提出质疑时，可发生争端解决。如果由接受方的付款指令规则集指定，则将自动退款从接受方发给发送方以解决争端。争端由发送方起始，由接受方所指定的争端解决策略管辖。在该例确认施例中，普通退款不会被质疑，而是可由接受方在任何时间请求并始终会成功，而与由这些规则指定的争端解决策略无关。

## “SENDER WINS”时间限定

```
duration SenderWinsTimeLimit := '15 days';
```

or, in alternative embodiments,

```
SenderWinsTimeLimit := duration 'P15D';
```

该规则指定如果发送方在交易的 15 天之内提出质疑，则将发出自动退款。

## 不自动退款

```
SenderWinsTimeLimit := '0 days';
```

或者,在可选实施例中

```
SenderWinsTimeLimit := duration 'P0D'
```

该规则指定如果发送方提出质疑则不会发出自动退款。

### “SENDER WINS”退款百分率或百分比限定

```
number SenderWinsRefundFraction := 50%;
```

或者,在可选实施例中

```
SenderWinsPercentLimit := 50%;
```

该规则指定仅 50% 的交易量会被退款以解决争端。

呼叫方/最终用户或付款接受方限定付款方法的示例如下:

呼叫方和/或接受方可设置有关资金出处的限定。例如,如果呼叫方正在支付交易费用,则呼叫方可能想确保资金的来源不是“信用卡”,因为信用卡交易费用高于其它方法的交易费用。

### 仅余额转帐

```
paymentMethod == 'balance transfer' ;
```

如果付款方法不是余额转帐,则该规则将禁止交易。

### 余额转帐和仅 ACH

```
paymentMethod == 'balance transfer' or paymentMethod == 'ach';
```

如果付款方法不是余额转帐或 ACH,则该规则将禁止交易。

### 无信用卡

```
paymentMethod != 'credit card' ;
```

如果付款方法是信用卡,则该规则将禁止交易。

限定付费百分率的示例如下:

费用百分率由呼叫方送入支付消息中。任何各方均可使用这些规则。

### 限定发送方的总费用百分率

SenderFractionOfFee <= 50%;

或者,在可选实施例中

SenderFeePercent <= 50%;

**如果发送方正支付多于 50% 的总付费, 则该规则将禁止交易。**

### 限定接受方的总费用百分率

RecipientFractionOfFee <= 80 %;

或者,在可选实施例中

recipientFeePercent <= 80%;

**如果接受方正支付多于 80% 的总付费, 则该规则将禁止交易。**

### 限定呼叫方的总费用百分率

callerFractionOf Fee <= 1/3;

或者,在可选实施例中

callerFeePercent <= 1/3;

**如果接受方正支付多于 1/3 的总付费, 则该规则将禁止交易。**

可用于付款指令规则集并且可作为支付请求消息的一部分被发送作为 XML 文件 (例如从呼叫方发送到交易授权方) 的上下文数据的示例如下:

参数	数值类型	说明
callerCredential	字符串	通过凭证服务创建并授权，用来确认呼叫方的身份
sendertoken	字符串	在其帐户中识别发送方的付款指令规则集
recipientToken	字符串	在其帐户中识别接受方的付款指令规则集
callerToken	字符串	在其帐户中识别呼叫方的付款指令规则集
senderFractionOfFees	数字	将由发送方支付的费用量的百分率
recipientFractionOfFees	数字	将由接受方支付的费用量的百分率
callerFractionOfFees	数字	将由呼叫方支付的费用量的百分率
transactionAmount	金额	在发送方和接受方之间的付款量
SenderTxnReference	字符串	对发送方有意义的交易的说明，在一些情况下其是可选的
RecipientTxnReference	字符串	对接受方有意义的交易的说明，在一些情况下其是可选的
CallerTxnReference	字符串	对呼叫方有意义的交易的说明，在一些情况下其是可选的
ServiceDescription	字符串	正提供的服务或产品（其中正进行付款）的说明，在一些情况下其是可选的

可用于权标的上下文数据的示例如下：

参数	数值类型	说明
TokenTransactionCount	数字	由该权标准许的交易数目，包括当前交易
TokenTransactionAmount	金额	由该权标准许的所有交易的总量，包括当前交易

使用付款指令规则集授权交易的付款的示例如下：

在该示例中，交易包括利用由发送方提供的应用程序请求来自接受方的 Web 服务的呼叫方。发送方支付与由交易授权方收费的请求的服务相关的费用，发送方支付交易的成本，并且交易量为 50 美元。作为对第三方交易授权方的交易授权呼叫的一部分的送入的参数值为：

- CallerFractionOfFees := 100%
- SenderFractionOfFees := 0%
- RecipientFractionOfFees := 0%
- TransactionAmount := USD 50.00

任何可选参数随交易被保存，但不影响执行交易。

### 数据检索和帐户验证

交易授权方接收来自呼叫方的交易授权调用，其包括用于呼叫方、接受方和发送方的权标，并检索与权标相关的数据。检索的数据如下：

- 每方的帐户信息，如对应的权标所识别的：
  - 帐户 ID
  - 状态 (活动的、停止的、挂起的)
  - 余额
  - 可用的余额

交易授权方验证各方的帐户是否存在以及是否均为活动的。如果任何帐户为不活动的，则禁止交易并返回错误。否则，交易授权方继续检索与付款指令规则集相关的数据：

- 付款指令规则集和权标每一方的统计：
  - 付款指令规则
  - 使用总数：迄今权标所允许的交易数
  - 使用总量：迄今权标所允许的交易量的和（绝对值）
  - 交易授权方本身的付款指令规则，用于指定费用、缺省和允许值。

下面将对每一权标示出并描述指令集。

#### 发送方：

# 每一交易限定为\$50

```
TransactionAmount <= 'USD 50';
```

# 限定权标使用为 10 及\$75 总量，无论哪个先出现

```
TokenTransactionCount <= 10;
```

```

TokenTransactionAmount <= 'USD 75';
# 权标在 200 年 1 月 1 日到期

TransactionDate <= '2004-Jun-1';
# 拒绝支付符合费用

senderFractionofFees == 0%;
# 从发送方的缺省 ACH 帐户提取资金

string PaymentMethod := 'ach';
# 争端解决: 接受方必须允许 90% 或以上 10 天或以上争端退款

SenderWinsTimeLimit >= '10 days';

senderwinsFractionLimit >= 90%;

```

#### 接受方:

```

# 权标在 2004 年 10 月 1 日到期

TransactionDate <= '2004-oct-1';
# 呼叫方必须支付所有费用

callerFractionOfFees == 100%;
# 争端解决: 允许 10 天或以上争端退款按 100%

duration SenderWinsTimeLimit := '10 days';

number SenderwinsFractionLimit := 100%;

```

#### 呼叫方:

# 如果付款方法不是 ACH 或余额转帐, 则交易失败

```
PaymentMethod in ( 'ach', 'balance transfer' );
```

#### 交易授权方:

# 如果付款方法不是 ACH、余额转帐或信用卡, 则交易失败

```
PaymentMethod in ( 'ach', 'balance transfer', 'credit card' );
```

交易授权方将检索的数据与在授权调用中支付请求消息中接收的数据合并, 并评估各方的付款指令规则集。

#### 评估付款指令规则集

下面将描述表示各付款指令规则集之间的交互的评估。这里所呈

现的付款指令规则集均适合。

发送方:

# 每一交易限定为\$50

TransactionAmount <= 'USD 50';

发送方正限定每一交易量为\$50。其不限制权标的总使用（见下一项）。

发送方:

# 限定权标使用为共计 10 次

MyTokenUseCount <= 10;

# 限定权标使用为共计\$75

MyTokenTotalAmount <= 'USD 75';

发送方限定权标的使用为共计 10 次或\$75，无论哪个先出现。

发送方:

# 权标在 2004 年 1 月 1 日到期

TransactionDate <= '2004-Jun-1';

接受方:

# 权标在 2004 年 10 月 1 日到期

TransactionDate <= '2004-Oct-1';

发送方的权标仅可使用到 2004 年 1 月 1 日，而接受方的权标可使用到 2004 年 10 月 1 日。

发送方:

# 拒绝支付任何费用

SenderFractionofFees == 0%;

接受方:

# 呼叫方必须支付所有费用

CallerFractionofFees == 100%;

发送方断言她将不支付任何费用。接受方更迫切地要求呼叫方支付所有费用。

发送方:

## # 从发送方的缺省 ACH 帐户提取资金

```
PaymentMethod := 'ach';
```

### 呼叫方:

# 如果付款方法不是 ACH 或余额转帐, 则交易失败。

```
PaymentMethod in ( 'ach', 'balance transfer' );
```

### 交易授权方:

# 如果付款方法不是 ACH、余额转帐或信用卡, 则交易失败。

```
PaymentMethod in ( 'ach', 'balance transfer', 'credit card' );
```

发送方将付款方法设置为 ACH, 这意味着资金将从与发送方的帐户连同交易授权方相关的缺省 ACH 银行帐户中被提取。由于呼叫方正支付 100% 的费用, 因此呼叫方断言付款方法必须为 ACH 或余额转帐, 以保持其成本下降。交易授权方的付款指令规则集确定付款方法被设定并限定其值。

### 接受方:

# 允许在交易日之后 10 天争端退款

```
duration senderWinsTimeLimit := '10 days';
```

# 允许 100% 争端退款

```
number senderwinsFractionLimit := 100%;
```

### 发送方:

# 接受方必须允许 10 天或以上争端退款

```
SenderWinsTimeLimit >='10 days';
```

# 接受方必须允许 90% 或以上争端退款

```
senderwinsRefundFraction >= 90%;
```

接受方指定争端解决策略, 其使发送方能够接收对交易日之后的 10 天 100% 的退款。发送方拒绝支付, 除非接受方指定她同意在交易日之后至少 10 天 90% 或以上的退款的争端解决策略。发送方和接受方达成一致。

## 执行交易



根据付款指令规则集的评估，可知下面的变量：

- PaymentMethod := 'ach'
- SenderWinsTimeLimit := 10 days
- SenderWinstFractionLimit := 100%

交易授权方搜寻发送方的缺省 ACH 金融工具以用作主交易的资金来源。根据费用清单，对于\$50的交易，ACH交易花费\$0.60（每一交易的1.0% + \$0.10）。交易授权方还生成将被返回给呼叫方的交易ID。基于所有这些，交易授权如下执行：

- 通过ACH从发送方的银行帐户提取\$50.00
  - 将\$50.00存入交易授权方的发送方的帐户中
  - 执行从发送方的交易授权方帐户到交易授权方的接受方的帐户余额转帐\$50.00
  - 执行从交易授权方的呼叫方的帐户到交易授权方余额转帐\$0.60
- 参考、描述以及争端解决变量均被存储在交易数据中。在执行上述交易期间，帐户余额被更新。

支付响应消息

由于交易是成功的，因此交易ID作为支付响应消息的一部分被返回给呼叫方。

参数	类型	说明
交易ID	字符串	交易的参考，可被提供给交易授权方用于查询、退款、争端及其它操作。

在一些实施例中，使用指令规则集被用来查看是否一些事件要被授权，并且包括对照上下文数据被评估的一组规则（也称作“断言表达式”或“断言”），---如果规则/断言对给定组的上下文数据评估为真，则事件被授权。下面是用于定义使用指令规则集的语言的示例：

断言示例：

在示例语言中，断言为评估为真或假的表达式（即为布尔表达式）。赋值也可被用于设置变量值。如果指令规则集中的所有断言评估为真，

则交易被允许，假如这样的话，如所赋值的变量值（如果存在的话）可被用于直接执行交易。

最简付款指令规则集由下列文本组成，如其始终评估为真：

```
true ;
```

请求对更多的断言添加逻辑操作、一些简化表达式和文字。看起来像是：

```
not false or false and true;
```

```
12.3 >= 7;
```

```
('12/05/2004' >= now) != ( TransactionAmount * 2 <= 'USD 4.10' );
```

上述付款指令规则集将始终评估为真（只要在 2004 年 12 月 5 日前）。

对于算符优先该示例语言中的表达式与 Perl 或 C 中的表达式类似。

为了说明，见以下表达式：

```
5 + 5 * 10 == 55 ;
```

上述评估为真，因为其等同于：

```
( 5 + ( 5 * 10 ) ) == 100 ;
```

通常，算符评估的优先按下列顺序：

1) Literals, blackboard dereferences, and expressions in parentheses ( )

2) negate - not !

3) times \* div /, left to right

4) minus - plus +, left to right

5) equals == notEquals != lessThan < greaterThan > lessThanOrEquals <= greaterThanOrEquals >=, in, not in

6) and &&, left to right

7) or ||, left to right

同样类似 C 或 Perl，在该示例语言中空白被忽略，因此制表符、空格和换行是无效的（除字符串引用符外，其中保留严格格式）。

## 评估示例

你可通过构成赋值语句创建变量并设置其值。赋值语句如下：

```
type Name := expression;
```

下面的语句创建称作 `MyFavoriteAmount` 的货币类型的变量，并且将值设置为 `USD 7.99`：

```
money MyFavoriteAmount := 'USD 7.99';
```

在付款指令规则集的评估期间变量值不变化。可有变量的多个赋值语句（即使在不同的付款指令规则集中），但所有值必须一致，否则其将按断言失败对待。变量被存储在黑板（`blackboard`）中，这意味着你可以在断言中去参考它们。在付款指令规则集（或同时评估的另一付款指令规则集）中评估的变量可被用于任何表达式中。

## 上下文黑板

用文字（`literal`）当表达式颇为无用，因为它们总是评估为同一结果。实际上，付款指令规则集针对一些组的上下文数据被评估，在该示例语言中其被存储在称作黑板（`blackboard`）的内存区中。在该示例中，黑板为标量值的字符串密钥的简单映射。字符串密钥通常以大写字母开始。名称区分大小写。黑板提供事件所特定的数据。

为了去参考表达式中的黑板数据，参考名称被简单地包括在表达式中：

```
TransactionAmount / 2 >= 'USD 5.00';
```

黑板去参考可被用于标量值可用的任何表达式中。

存储在黑板中的变量为全局的或专用的。专用变量可仅由特定付款指令规则集（例如发送方、接受方或呼叫方）使用，而全局变量可由任何付款指令规则集用于表达式中。

在该示例中，变量被赋予不变化的单一值。对于其值，不同的变量具有不同的源。例如，一些变量在付款指令规则集内被赋予其值，一些变量由交易授权系统（“TAS”）指定，一些变量由呼叫方在支付请求消息中送入。

下面的图表表示对各支付请求消息定义的变量，以及每一变量值的源。此外，付款指令规则集可利用赋值语句创建其自己的变量。这些变量为全局变量；例如，由发送方的付款指令规则集中的赋值语句创建的变量可被用于接受方指令的表达式中。用户指定的变量名应该对用户特定的字符的识别次序开始，以避免与 TAS 定义的变量名和其它用户定义的变量冲突。例如，由 Bob' Surf Shack 定义的变量名可均由“BobsSurfShack”开始。

有几个例外，所有全局变量可由呼叫方在支付请求消息中送入，或者可在任何付款指令规则集中指定，因此，下面的图表中所示的当前源仅为惯例。其例外情况包括具有由 TAS 指定的值的变量，事实是，至少一个权标必须送入到来的授权消息，并且要使用的任何其它权标必须为可获得的。例如，如果呼叫方权标 (SenderToken) 仅为付款请求消息中提供的权标，则对应的付款指令规则集可指示发送方权标或接受方权标 (RecipientToken) 或这两者。如果其仅指示发送方权标，则发送方的付款指令规则集可指示接受方权标，反之亦然。

## 全局变量

变量名	来源	类型	说明
TransactionTimestamp	由 TAS 指定	数据日期	交易的数据和日期
TransactionID	由 TAS 指定	字符串	交易的 ID
TotalFeeAmount	由 TAS 指定	金额	该交易的支付给 TAS 的费用总量
SenderFractionOfFees	由呼叫方在支付请求消息中送入	数字	发送方将支付的 TAS 费用的百分率, 值必须在 0 和 1 之间
RecipientFractionOfFees	由呼叫方在支付请求消息中送入	数字	接收方将支付的 TAS 费用的百分率, 值在 0 和 1 之间
CallerFractionOfFees	由呼叫方在支付请求消息中送入	数字	呼叫方将支付的 TAS 费用的百分率, 值在 0 和 1 之间
TransactionAmount	由呼叫方在支付请求消息中送入	金额	交易量
TransactionDescription	由呼叫方在支付请求消息中送入	字符串	什么正被购买的人可读说明。随交易存储并当显示该交易用于报告或查帐时使用。
TransactionData	由呼叫方在支付请求消息中送入	字符串	由关于该交易的呼叫方提供的数据。例如, 呼叫方可存储正出售服务的 XML 说明。该数据随交易被存储但对 TAS 不透明
SenderToken	由呼叫方在支付请求消息中送入	字符串	涉及发送方的付款指令规则集的权标
RecipientToken	由呼叫方在支付请求消息中送入	字符串	涉及接受方的付款指令规则集的权标
CallerToken	由呼叫方在支付请求消息中送入	字符串	涉及呼叫方的付款指令规则集的权标
PaymentMethod	由发送方的付款指令规则集指定	字符串	指定用作资金源的金融工具的类型: ACH、信用卡、转帐帐户之一。
PaymentInstrument	由发送方的付款指令规则集指定	字符串	指定用作资金源的金融工具的参考
SenderWinsTimeLimit	由接受方的付款指令规则集指定	持续时间	指定在交易时间戳后发送方可通过提起争端自动获取退款的最大时间。退款量由 SenderWinsRefundFraction 限定
SenderWinsRefundFraction	由接受方的付款指令规则集指定	数字	指定发送方可通过提起争端自动获取退款的交易量的百分率。争端必须在由 SenderWinsTimeLimit 指定的时间内被提起

## 专用变量

在下面图表中列出的专用变量被提供给每一付款指令规则集，以使每一付款指令规则集得到这些变量的每一个的其自身的拷贝。例如，这些参考中的数据对发送方和对接受方不同。接受方不能访问发送方的值，反之亦然。

变量名	来源	类型	说明
MyTokenUseCount	由 TAS 指定	数字	由该权标批准的交易数，优选包括当前交易
MyTokenUseTotalAmount	由 TAS 指定	金额	由该权标批准的所有交易的总量，优选包括当前交易
MyFeeAmount	由 TAS 指定	金额	由该付款指令规则集支付的费用量。例如，对呼叫方的指令，其等于 <b>CallerFractionOfFeestimes TotalFeeAmount</b>
MyRole	由 TAS 指定	字符串	交易中该权标被调用的一方的作用。发送方、接受方、呼叫方之一。
MyTokenID	在付款指令规则集中指定或被送入以安装付款指令规则集消息	字符串	用于付款指令规则集的对用户有意义的 ID
MyTokenDescription	在付款指令规则集中指定或被送入以安装付款指令规则集消息	字符串	付款指令规则集的对用户有意义的人可读说明
MyTokenData	在付款指令规则集中指定或被送入以安装付款指令规则集消息	字符串	与付款指令规则集相关的对用户有意义的机器可读数据
MyTransactionID	发送方、接受方和呼叫方值，均由呼叫方在支付请求消息中送入	字符串	用于交易的 ID，其仅对发送方、接受方或呼叫方有意义。随交易存储。
MyTransactionDescription	发送方、接受方和呼叫方值，均由呼叫方在支付请求消息中送入	字符串	什么正被购买的人可读说明，其仅对发送方、接受方或呼叫方有意义。随交易存储。
MyTransactionData	发送方、接受方和呼叫方值，均由呼叫方在支付请求消息中送入	字符串	由关于该交易的呼叫方提供的数据，其仅对发送方、接受方或呼叫方有意义。例如，接受方可存储正出售服务的 XML 说明。该数据随交易被存储，但其对 TAS 不透明

## 注释

注释以磅字符(#)开始，并且使该行的其余部分被忽略。对于多行注释，在每行的开始放置#。

```
# this is a comment that
```

```
# spans more than one line
```

## 语言特性

尽管在一些实施例中语言可包括诸如循环结构和/或递归之类的时间方面，但在本示例中，不存在这样的时间方面或分支语句。数据类型包括布尔、数字、字符串、日期时间、持续时间和金额。算符如下：

=	等于：确认是否左右侧相等
!=	不等于：确认是否左右侧不相等
<	小于
>	大于
<=	小于或等于
>=	大于或等于
<=	小于或等于
:=	赋值：将左侧值赋值到右侧
+	加
-	一元求反或减
*	乘
/	除
&&	与
	或
!	非

## 文字

### 布尔

真

假

### 数字

123

0.234

5%

允许直到  $2^{31} - 1$  的任何整数。负值通过“求反”操作实现。也支持 10 进制数，并且不被存储为浮点数。数以阿拉伯数字开始，因此前导零被用于零和壹之间的小数。还支持将数字按百分比显示。

### 字符串

'any text'

字符文字被置于单引号中。通常不使用双引号。

### 日期时间

now

'YYYY-MON-DD'

'YYYY-MON-DD HH : MM : SS AM'

“now”文字返回当前时间。紧接该图案的引号内的文本可被比较。日期时间关键字随引号内的字符串而不随字符串表达式工作。

### 持续时间

'10 days'

持续时间文字指定时间段，并且可被增加日期以形成新日期。各种数学运算是可用的。

'1 year 2 months 3 days 10 hours 30 minutes'

'P1Y2M3DT10H30M'

上面两种文字格式均被支持。第二种格式翻译成 1 年 2 月 3 日 10 时 30 分。持续时间文字的第二种形式遵循来自 W3C XML Schema Specification 的持续时间的定义。根据 [http: /](http://)



**/www.w3.org/TR/xmlschema-2/#duration:**

持续时间的词汇表示法为[ISO 8601] 扩展格式 **PnYnMnDTnHnMnS**，其中 **nY** 表示年数，**nM** 表示月数，**nD** 表示日期数，**T** 为日期/时间分隔符，**nH** 表示时数，**nM** 表示分数，**nS** 表示秒数。

**金额**

```
'USD 1.00'
```

```
money ( 'USD', TokenTransactionCount * 0.05 )
```

金额文字可由货币代码和数字文字或货币代码和返回数字的表达式指定。有关货币代码的更多信息见 <http://www.bsi-global.com/iso4217currency>，包括列出所有货币代码的表格。允许的货币代码的设置不限于由 ISO4217 指定的这些，ISO 4217 货币代码均可被承认。

**操作和内置功能**

在以下部分中，将分别说明代表左手侧操作数和右手侧操作数的 **lhs** 和 **rhs**

**negate 操作**

```
- rhs
```

```
negate rhs
```

允许类型：整数、双精度型、持续时间、金额

**not 操作**

```
! rhs
```

```
not rhs
```

允许类型：布尔型

**or 操作**

```
lhs or rhs
```

```
lhs | rhs
```

允许类型：布尔型

**and 操作**

lhs and rhs

lhs && rhs

**允许类型：布尔型**

### **in 操作**

lhs in ( rhs1, rhs2, rhs3... )

**允许类型：所有类型**

**lhs 和 rhs 应为同一类型，除非整数型和双精度型可比较。in 操作为简写形式，完全等同于以下：**

lhs == rhs1 or lhs == rhs2 or lhs == rhs 3 ...

### **notin 操作**

lhs notIn ( rhs1, rhs2, rhs3 ... )

**允许类型：所有类型**

**lhs 和 rhs 应为同一类型，除非整数型和双精度型可比较。notin 操作作为简写形式，完全等同于以下：**

lhs != rhs1 and lhs != rhs2 and lhs != rhs3 ...

### **equals 操作**

lhs equals rhs

lhs == rhs

**允许类型：所有类型**

**lhs 和 rhs 应为同一类型，除非整数型和双精度型可比较。**

### **notequals 操作**

lhs notEquals rhs

lhs != rhs

**允许类型：所有类型**

**lhs 和 rhs 应为同一类型，除非整数型和双精度型可比较。**

### **lessthan 操作**

lhs lessThan rhs

lhs < rhs

**允许类型：整数、双精度型、金额、日期时间、持续时间。**

**lhs 和 rhs 应为同一类型，除非整数型和双精度型可比较。**

### **greaterthan 操作**

lhs greaterThan rhs

lhs > rhs

**允许类型：整数、双精度型、金额、日期时间、持续时间。**

**lhs 和 rhs 应为同一类型，除非整数型和双精度型可比较。**

### **lessthanorequals 操作**

lhs lessThanOrEquals rhs

lhs >= rhs

**允许类型：整数、双精度型、金额、日期时间、持续时间。**

**lhs 和 rhs 应为同一类型，除非整数型和双精度型可比较。**

### **greaterthanequals 操作**

lhs greaterThanorEquals rhs

lhs >= rhs

**允许类型：整数、双精度型、金额、日期时间、持续时间。**

**lhs 和 rhs 应为同一类型，除非整数型和双精度型可比较。**

### **add 操作**

lhs plus rhs

lhs + rhs

**允许类型：整数、双精度型、金额、日期时间、持续时间。**

**整数和双精度型可被操作，其将产生双精度型，除非两个操作数均为整数。**

**有关金额的操作需要 lhs 和 rhs 均为金额类型。**

**持续时间可与持续时间相加以产生持续时间。**

**日期时间可与持续时间相加或持续时间可与日期时间相加以产生日期时间。**

### **SUB 操作**

lhs minus rhs

lhs - rhs

**允许类型：整数、双精度型、金额、日期时间、持续时间。**

整数和双精度型可被操作，其将产生双精度型，除非两个操作数均为整数。

有关金额的操作需要 lhs 和 rhs 均为金额类型。

持续时间可与持续时间相减以产生持续时间。

持续时间可与日期时间相减，日期时间必须为 lhs，持续时间必须为 rhs，从而产生日期时间。

### MULT 操作

lhs times rhs

lhs \* rhs

**允许类型：整数、双精度型、金额、持续时间**

整数和双精度型可被操作，其将产生双精度型，除非两个操作数均为整数。

对于金额，金额可乘以整数或双精度型以产生金额。

对于持续时间，持续时间可乘以整数以产生持续时间。

### DIV 操作

lhs div rhs

lhs / rhs

**允许类型：整数、双精度型、金额**

整数和双精度型可被操作，其将产生双精度型，除非两个操作数均为整数。

对于金额，金额可除以整数或双精度型，lhs 必须为金额值并且 rhs 必须为整数或双精度型，产生金额。

### MIN 功能

min( lhs, rhs )

**允许类型：整数、双精度型、金额、日期时间、持续时间**

lhs 和 rhs 应为同一类型，除非整数和双精度型可被比较。

### MAX 功能

max( lhs, rhs )

**允许类型：整数、双精度型、金额、日期时间、持续时间**  
**lhs 和 rhs 应为同一类型，除非整数和双精度型可被比较。**

### **MOD 功能**

`mod( lhs, rhs )`

**允许类型：整数**

**产生 lhs 除 rhs 的整数余数。**

### **CAT 功能**

`cat ( lhs, rhs )`

**允许类型：字符串**

**产生 lhs 和 rhs 的组合字符串。**

### **GETCURRENCYCODE**

`getCurrencycode ( rhs )`

**允许类型：金额**

**返回货币代码作为字符串。**

**语法分析**

**Tokens:**

```

OR = "||";
AND = "&&";
NOT = "!";
EQUALS = "==" ;
NOT_EQUALS = "!=";
L_THAN = "<";
G_THAN = ">";
LT_EQ = "<=";
GT_EQ = ">=";
PLUS = "+";
MINUS = "-";
ASTERIK = "*";
DIV = "/";
ASSIGN = ":=";
ASSERTION;

```

```

/////////////////////////////////////////////////////////////////
// Parser Rules

```

**instructionSet**

```

: ( statement )+
;

```

**statement**

```

: ( "duration"
  | "money"
  | "datetime"
  | "string"
  | "number"
  )?
  BB_REFERENCE ASSIGN
  expression STATEMENT_END

| exp:expression STATEMENT_END

```

```

;

// expressions

expression
: andExp ( ( OR | "or" ) andExp )*
;

andExp
: compareExp ( ( AND | "and" ) compareExp )*
;

compareExp
: addExp
  ( ( EQUALS | "equals"
    | NOT_EQUALS | "notEquals"
    | L_THAN | "lessThan"
    | G_THAN | "greaterThan"
    | LT_EQ | "lessThanOrEquals"
    | GT_EQ | "greaterThanOrEquals"
    ) addExp
  | ( "in" | "notIn" )
    LPAREN expression ( COMMA expression )* RPAREN
  )
;

addExp
: multExp
  ( ( PLUS | "plus"
    | MINUS | "minus"
    ) multExp )*
;

multExp
: unaryExp
  ( ( ASTERIK | "times"
    | DIV | "div"
    ) unaryExp )*
;

unaryExp
: ( MINUS | "negate"
  | NOT | "not"
  ) scalarExp
| scalarExp
;

```

```
scalarExp
  : literal
  | function
  | BB_REFERENCE
  | LPAREN expression RPAREN
  ;

// functions

function
  : minFunction
  | maxFunction
  | catFunction
  | getCurrencyCodeFunction
  ;

minFunction
  : "min" LPAREN expression ( COMMA expression )+ RPAREN
  ;

maxFunction
  : "max" LPAREN expression ( COMMA expression )+ RPAREN
  ;

catFunction
  : "cat" LPAREN expression ( COMMA expression )+ RPAREN
  ;

getCurrencyCodeFunction
  : "getCurrencyCode" LPAREN expression RPAREN
  ;

// literals

literal
  : booleanLiteral
  | numberLiteral
  | dateTimeLiteral
  | durationLiteral
  | stringLiteral
  | moneyLiteral
  ;

booleanLiteral
  : "true"
  | "false"
  ;
```



```

moneyLiteral
  : "money" LPAREN QUOTED_LITERAL COMMA expression RPAREN
  ;

numberLiteral
  : NUMBER_LITERAL ( PERCENT )?
  ;

dateTimeLiteral
  : "dateTime" LPAREN expression RPAREN
  | "now"
  ;

durationLiteral
  : "duration" LPAREN expression RPAREN
  ;

stringLiteral
  : QUOTED_LITERAL
  ;

```

## 词汇定义

```

// character classifications
DOT          : '.';
LPAREN       : '(';
RPAREN       : ')';
COMMA        : ',';
STATEMENT_END : ';';
PERCENT      : '%';

protected QUOTE_CHAR : '\\';
protected DIGIT_CHAR : ( '0'..'9' );
protected ALPHA_CHAR : ( 'a'..'z' | 'A'..'Z' );
protected NAME_CHAR  : ( ALPHA_CHAR | DIGIT_CHAR | '_' );

protected OPERATOR_CHAR
  : ( '|' | '&' | '!' | '<' | '>' | '=' | '+' | '-' | '*' | '/' | ':' )
  ;

// skip all whitespace
WHITESPACE
  : ( ' '
    | '\t'
    | '\n'

```

```
        | '\r'
    )+
;

// token classifications
BB_REFERENCE
    : ALPHA_CHAR ( NAME_CHAR )*
;

NUMBER_LITERAL
    : ( DIGIT_CHAR )+ ( '.' ( DIGIT_CHAR )+ )?
;

QUOTED_LITERAL
    : QUOTE_CHAR
      ( ~( '\n' | '\'' )
      | '\n'
      )+
      QUOTE_CHAR
;

OPERATOR
    : OPERATOR_CHAR ( OPERATOR_CHAR )?
;

COMMENT
    : '#' ~( '\n' | '\r' )
      ( '\n'
      | '\r' )+
;
;
```

从上述可见，尽管出于说明目的已描述了特定实施例，但在不脱离本发明的精神和范围的情况下可进行各种修改。因此，除后附的权利要求和其中所述部分外，本发明不受限定。此外，尽管本发明的某些方面以某些权利要求形式在下面被呈现，但发明人旨在以任何可获得的权利要求形式呈现本发明的各个方面。例如，尽管当前仅本发明的一些方面在计算机可读介质中被具体化描述，但其它方面也同样被这样具体化。

由Web服务提供方ABC交互创建用户帐户的示例

欢迎, Web服务提供方ABC! 请在下面设置信息以创建您的用于WS付款处理系统(“PHS”)的帐户

1. 指定你的帐户名称和密码

名称:  密码:

可选的组织关系:

可选的证书:

2. 可选地指定你接收付款的一个或多个收支库  
你的PHS帐户(名为“MyPHSAccount”)也可被用作收支库

库名称及类型	库号及截止日期	限制
ABC-Biz   bank account   ▼	123456789-123-456789   <none>	<none>   ▼
PerDCard   debit card   ▼	1234-5678-1234-5678   12/06	min. pay.   ▼   \$0.05
<input type="text"/>   <none>   ▼	<input type="text"/>	<none>   ▼

3. 可选地指定你进行付款的一个或多个支付工具  
你的PHS帐户也可以被用作支付工具, 上限为\$1000.

库名称及类型	库号及截止日期	限制
ABC-Biz   bank account   ▼	123456789-123-456789   <none>	max. pay.   ▼   \$50.
PerCCard   credit card   ▼	1234-5678-5678-1234   01/06	max. bal.   ▼   \$2000.
<input type="text"/>   <none>   ▼	<input type="text"/>	AND   ▼   max. pay.   ▼   \$200.
<input type="text"/>   <none>   ▼	<input type="text"/>	<none>   ▼

116a  116b

图1A

由Web服务提供方用户ABC交互创建使用  
指令规则集的例子

欢迎，Web服务提供方ABC！请在下面设置信息以创建  
用于你的WSP-ABC帐户的使用指令规则集

117

使用指令规则集名称：  
用作： 收支库：  
截止日期：

指定该使用指令规则集的一个或多个规则

规则属性	标准	值
<input type="text" value="AllowedPayers"/>	<input type="text" value="Payer.OrganizationAffiliation"/> IS	<input type="text" value="Company XYZ"/>
<input type="text" value="OR"/>	<input type="text" value="Payer.Rating"/>	<input type="text" value="IS BCDCorp:Trusted"/>
<input type="text" value="OR"/>	<input type="text" value="Payer.ID"/>	<input type="text" value="IS 4215345829"/>
<input type="text" value="DisputeResolutionPolicy"/>	<input type="text" value="exclusive remedy is refund"/>	

118

<input type="text" value="AllowedPayers"/>	(default value: <all>).
<input type="text" value="DisallowedPayers"/>	(default value: <none>)
<input type="text" value="AllowedTransactionTypes"/>	(default value: <all>)
<input type="text" value="DisallowedTransactionTypes"/>	(default value: <none>)
<input type="text" value="DisputeResolutionPolicy"/>	(default value: <any>)
<input type="text" value="DisputeTimeWindow"/>	(default value: 30 days).
<input type="text" value="AllowedPaymentInstrumentTypes"/>	(default value: <all>)
<input type="text" value="DisallowedPaymentInstrumentTypes"/>	(default value: <none>)
<input type="text" value="AllowedCurrencies"/>	(default value: US dollars)
<input type="text" value="DisallowedCurrencies"/>	(default value: <all>)
⋮	
--OR--	
<User-Defined-Rule ("UDR")>	

119

图 1B

由应用开发方用户CDE使用XML文件程序性  
创建使用指令规则集的示例

```

<?xml version="1.0" ?>
<!DOCTYPE usage-instruction-rule-set >

<USAGE-INSTRUCTION-RULE-SET>
  <ACCOUNT-ID>1348193287</ACCOUNT-ID>
  <UIRS-NAME>MyPremiumCommercialApplications</UIRS-NAME>
  <UIRS-USAGE>payment provider</UIRS-USAGE>
  <UIRS-PAYMENT-INSTRUMENT>MyPHSAccount</UIRS-PAYMENT-INSTRUMENT>
  <UIRS-EXPIRATION-DATE>01-01-2005</UIRS-EXPIRATION-DATE>
  <UIRS-RULES>
    <RULE>
      <RULE-ATTRIBUTE>Recipient</RULE-ATTRIBUTE>
      <RULE-CRITERIA>Recipient.ID</RULE-CRITERIA>
      <RULE-EVALUATION>IS-NOT</RULE-EVALUATION>
      <RULE-VALUE>6242453245</RULE-VALUE>
    </RULE>
    <RULE>
      <RULE-ATTRIBUTE>PaymentsAllowed</RULE-ATTRIBUTE>
      <RULE-EVALUATION>LESS-OR-EQUAL</RULE-EVALUATION>
      <RULE-VALUE>$1000</RULE-VALUE>
    </RULE>
  </UIRS-RULES>
</USAGE-INSTRUCTION-RULE-SET>

```

131

---

```

<?xml version="1.0" ?>
<!DOCTYPE usage-instruction-rule-set >

<USAGE-INSTRUCTION-RULE-SET>
  <ACCOUNT-ID>1348193287</ACCOUNT-ID>
  <UIRS-NAME>SharewareApplicationAAA</UIRS-NAME>
  <UIRS-USAGE>access-authorizer</UIRS-USAGE>
  <UIRS-ACCESS-INFORMATION>
    <USERNAME>CDE-Company</USERNAME>
    <PASSWORD>aldkf32aM</PASSWORD>
  </UIRS-ACCESS-INFORMATION>
  <UIRS-RULES>
    <RULE>
      <RULE-ATTRIBUTE>User-Defined-Rule</RULE-ATTRIBUTE>
      <RULE-UDR-NAME>Allowed-Use</RULE-UDR-NAME>
      <RULE-INVOCATION>
        http://verify.cde.com/query?app=AAA&destination=[access-provider.name]
      </RULE-INVOCATION >
    </RULE>
  </UIRS-RULES>
</USAGE-INSTRUCTION-RULE-SET>

```

133

图 1C

WS付款处理系统数据库的示例

帐户DB

141	142	143	144	145	146	147	148	149	150
帐户ID	用户ID	名称	密码	组织关系	证书	收支库ID	支付工具ID	当前帐户余额	最大帐户余额
3419872987	0819343244	WSP-ABC	q3adouRT	ABC; Certified Web Service Provider Association	BCDCorp: Trusted	3419811; 3204980	3419821; 3947429	-	\$1000.
1348193287	-	CDE-Corp	-	-	BCDCorp: Trusted	-	-	\$231.25	\$500.

收支库DB

181	182	183	184	185	186	187	188
收支库ID	帐户ID	名称	类型	名称	截止日期	限制类型	限制值
3419811	3419872987	ABC-Biz	bank account	123456789-123-456789	-	-	-
3204980	3419872987	PerDCard	debit card	1234-5678-1234-5678	12/96	min. pay	\$0.05

支付工具DB

191	192	193	194	195	196	197	198
支付工具ID	帐户ID	名称	类型	名称	截止日期	限制类型	限制值
3419821	3419872987	ABC-Biz	bank account	123456789-123-456789	-	max. pay.	\$50.
3947429	3419872987	PerCCard	credit card	1234-5678-5678-1234	01/96	max. bal.;	\$2000.;
7982348	8049582022	CCI	credit card	0984-4328-2452-6754	03/98	max. pay.	\$200.
						max. bal.	\$10000.

图1D

使用指令规则集DB

161 UIRS ID	162 帐户ID	163 使用权标	164 名称	165 使用类型	166 收支库/ 支付工具	167 截止日期	168 规则ID
243919845	3419872987	84331984091340	TrustedCustomerOfMyMapService	payment-recipient	3204980	.	0298452; 2343248
341290809	1348193287	74993279274592	MyPremiumCommercialApplications	payment-provider	1348193287	01/01/2005	0984970; 2934701
::							

使用指令规则DB

171 规则ID	172 UIRS ID	173 规则属性	174 标准	175 评估	176 值	177 接续
0298452	243919845	Payer	Payer.Organization.Affiliation; Payer.Rating; Payer.ID	IS; IS; IS;	Company XYZ, BCDCorp:Trusted; 4215345829	OR; OR
2343248	243919845	DisputeResolutionPolicy		IS;	exclusive remedy is refund	-
0984970	341290809	Recipient	Recipient.ID	IS-NOT; LESS-OR- EQUAL;	6242453245 \$1000.	- -
2934701	341290809	PaymentsAllowed				-
::						

图1E

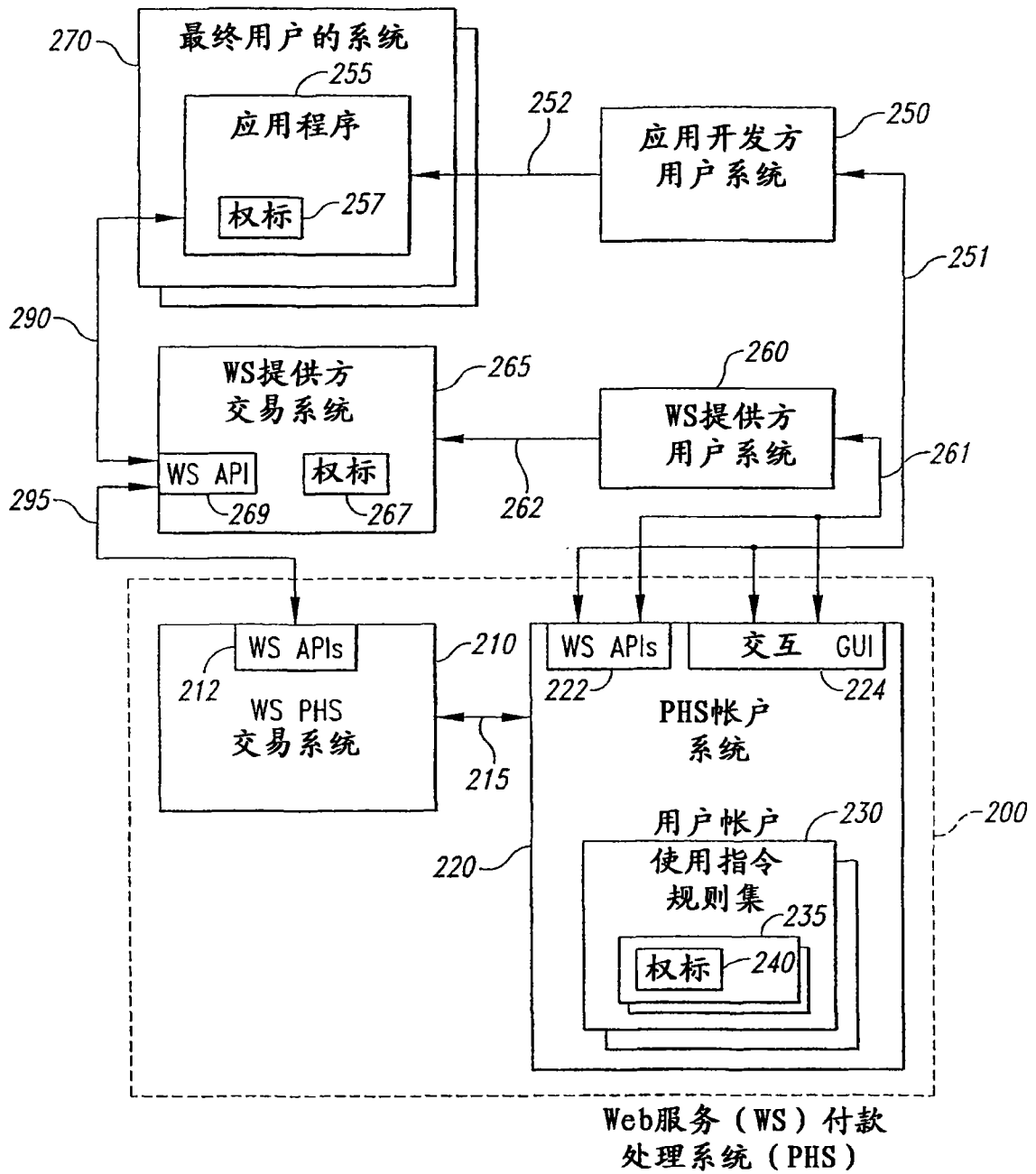


图2



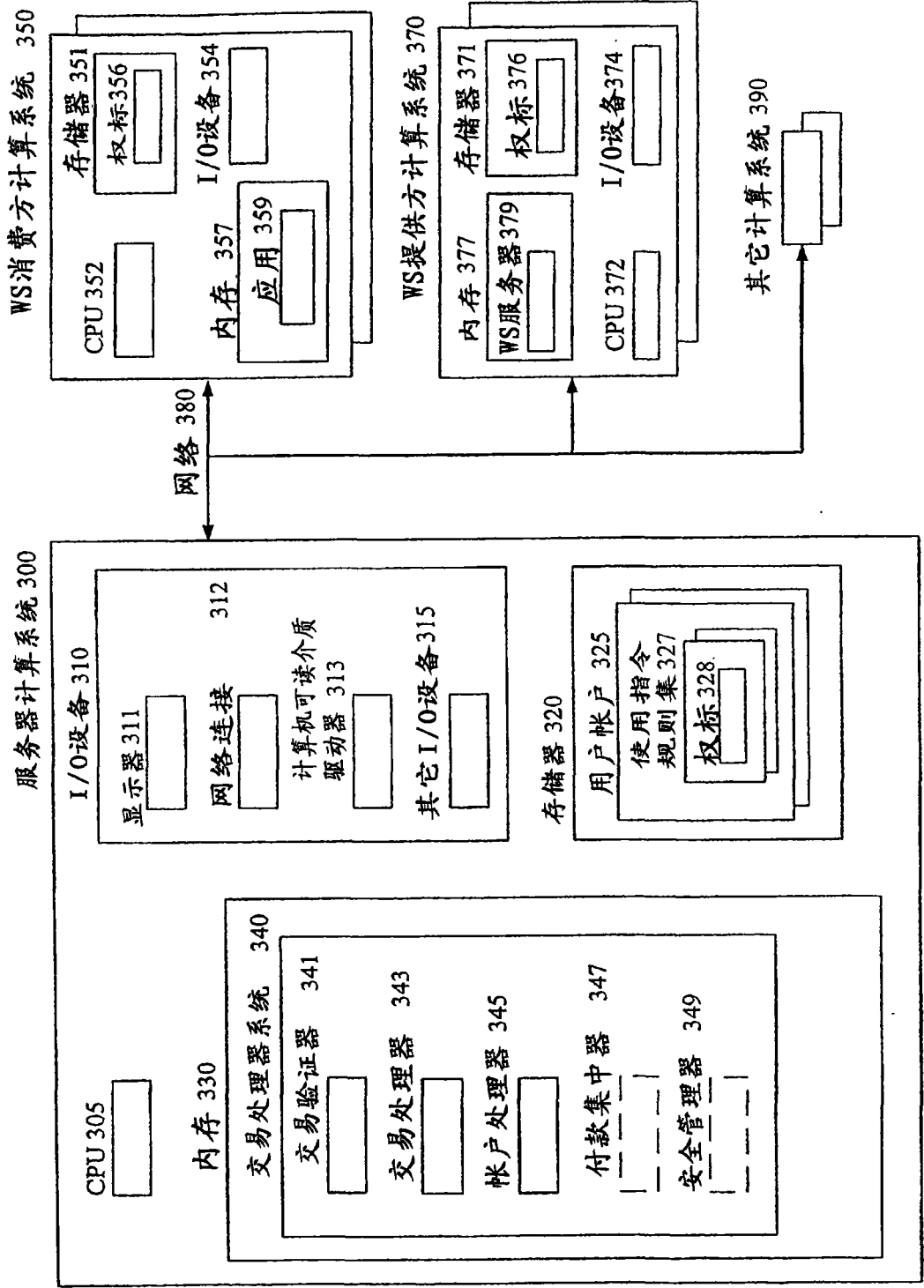


图 3

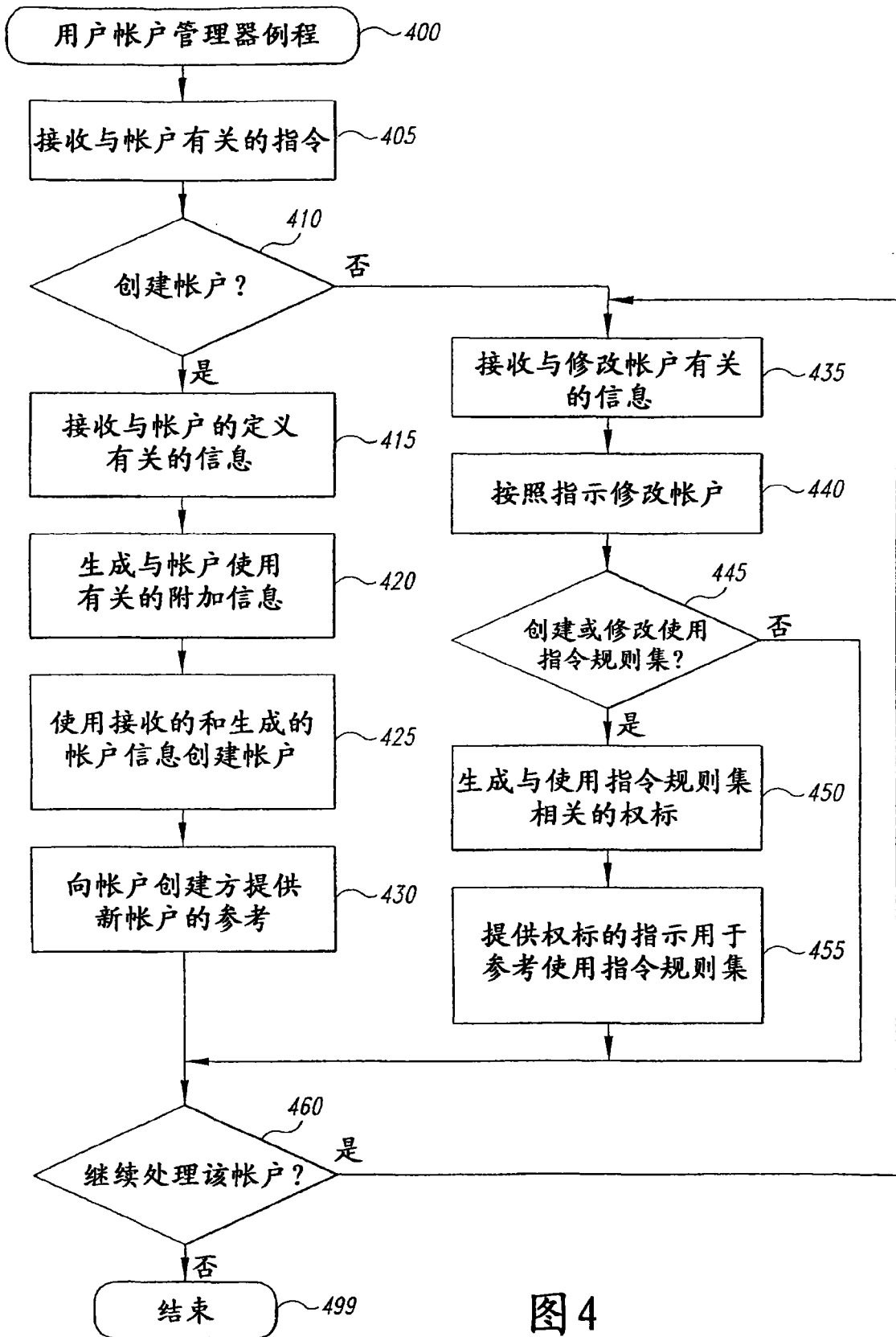


图4

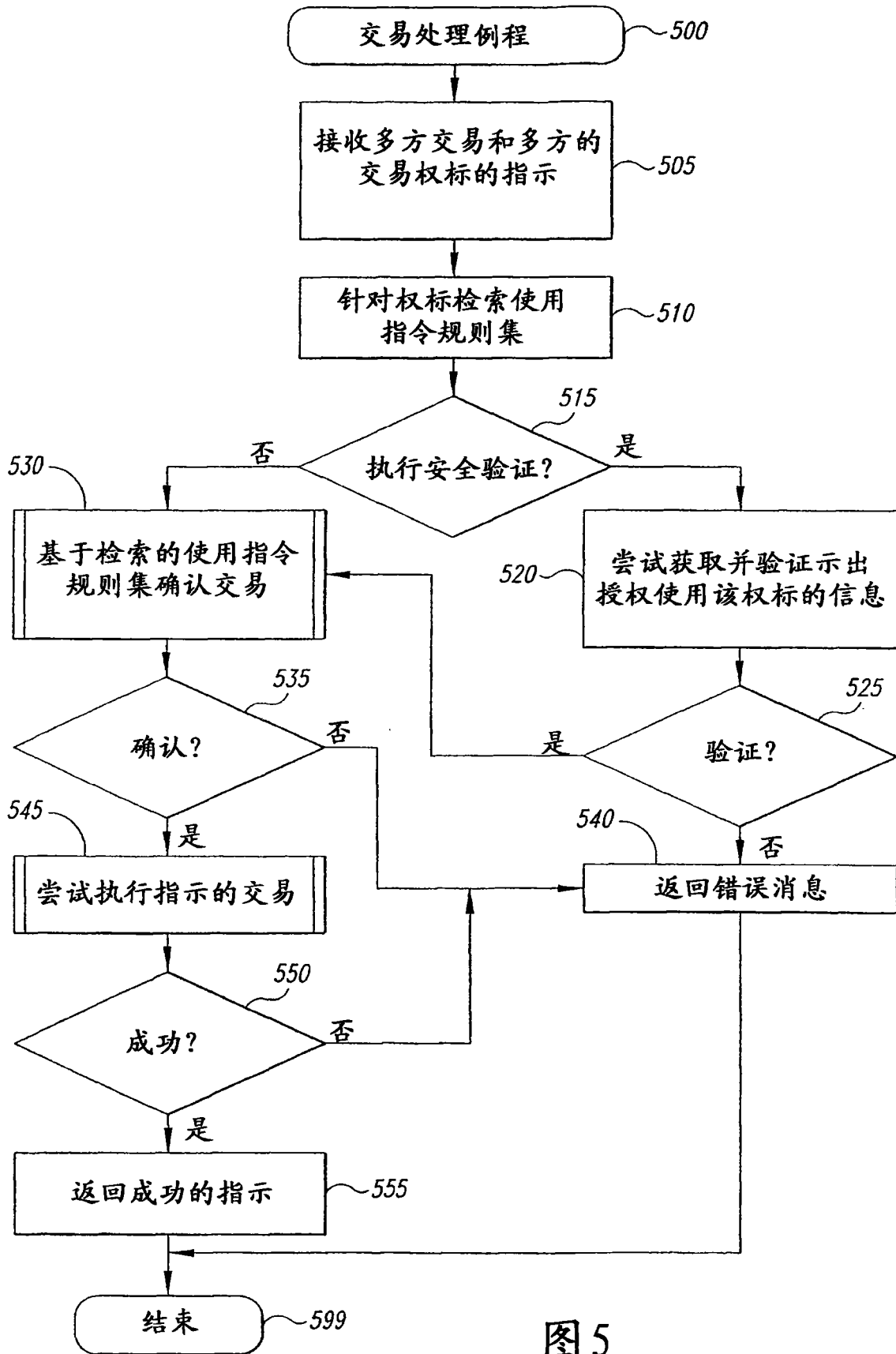


图 5

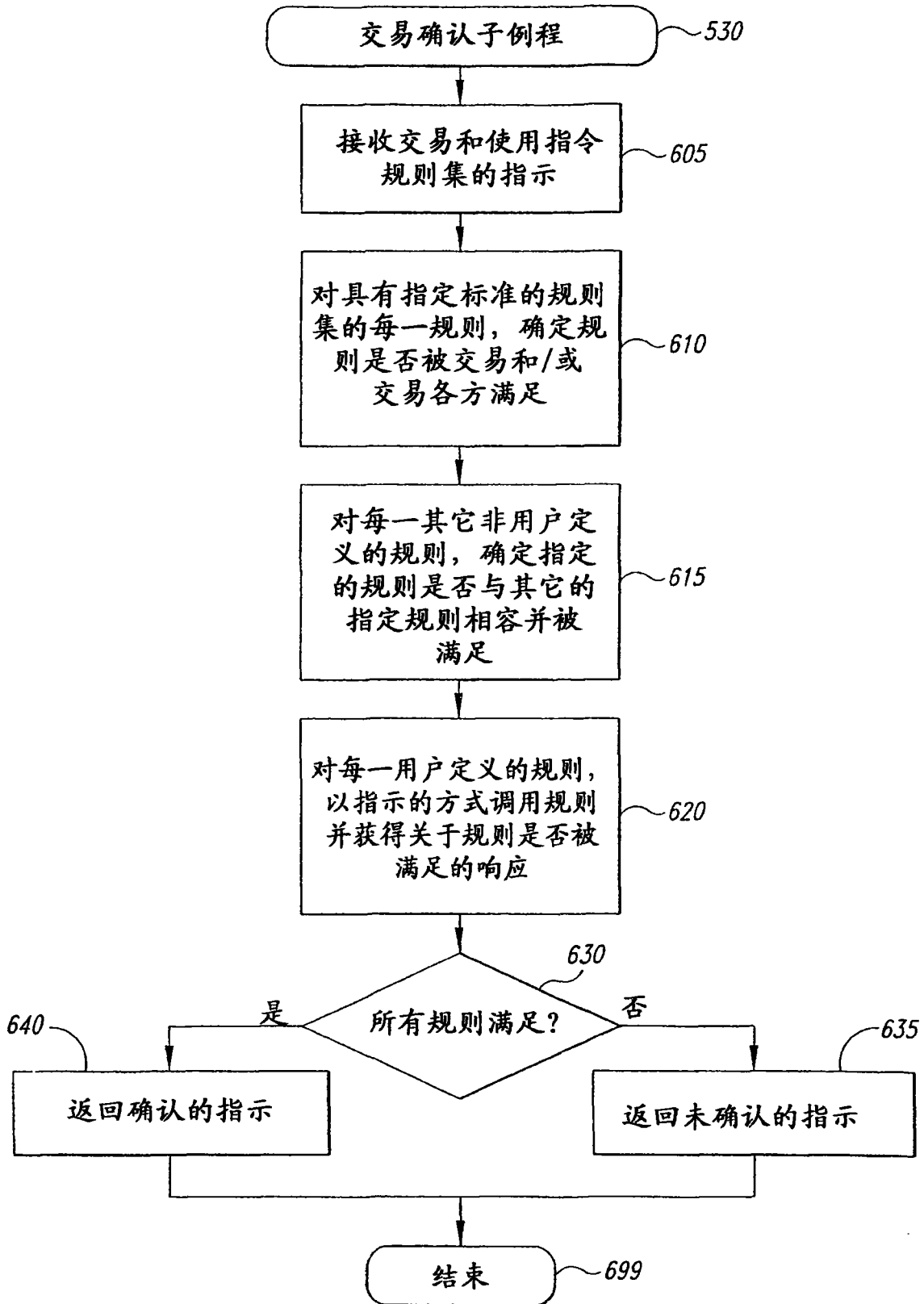
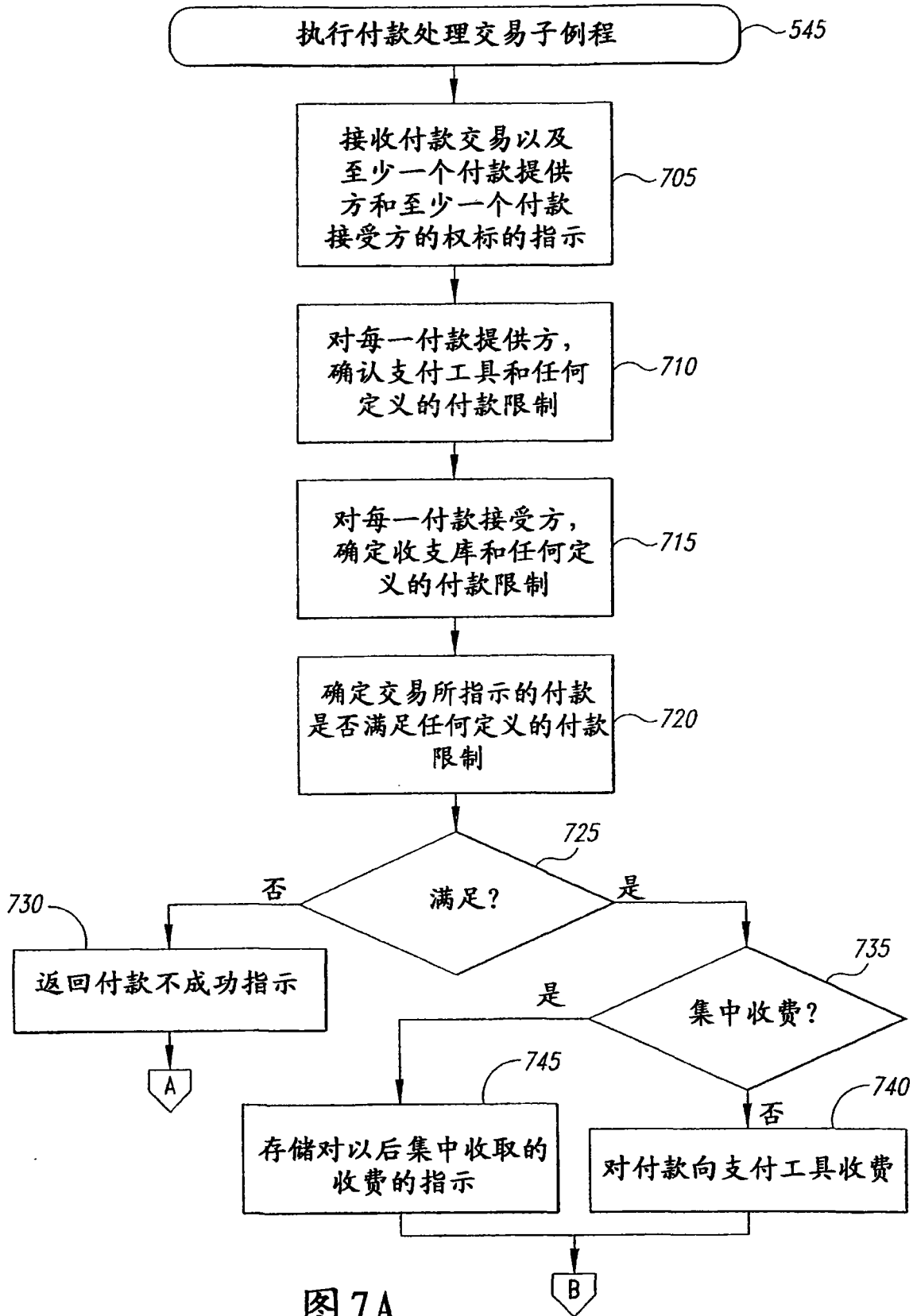


图6



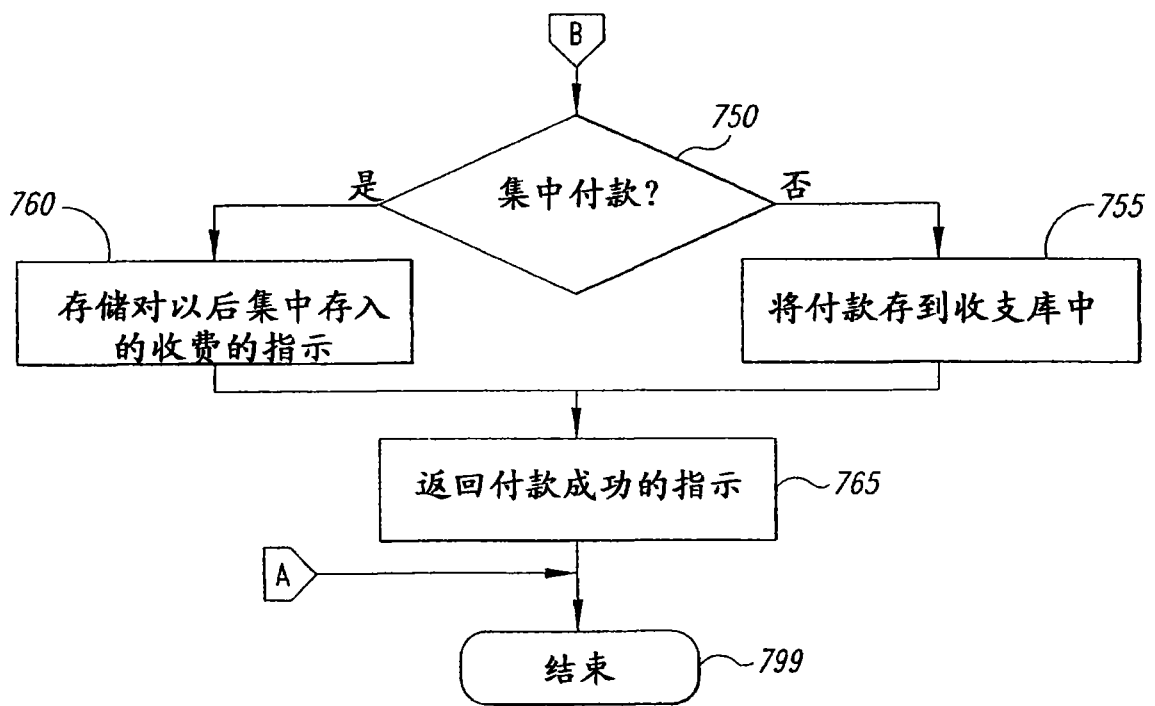


图7B

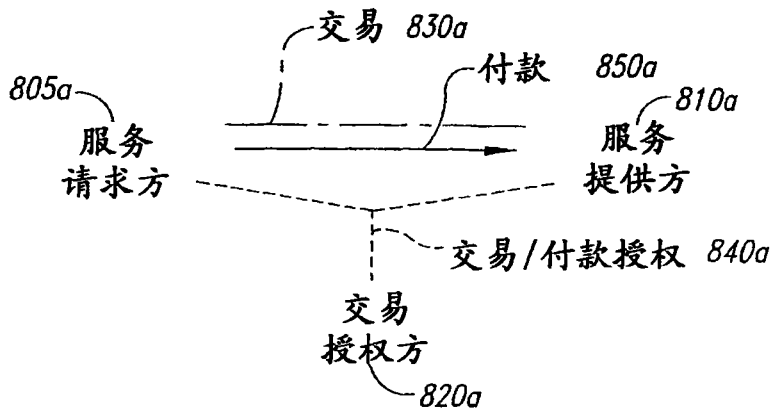


图 8A

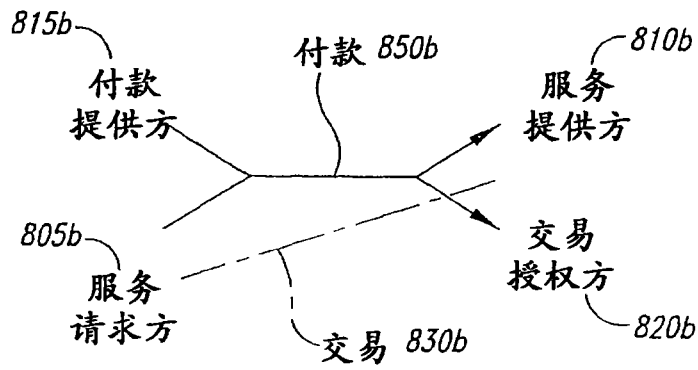


图 8B

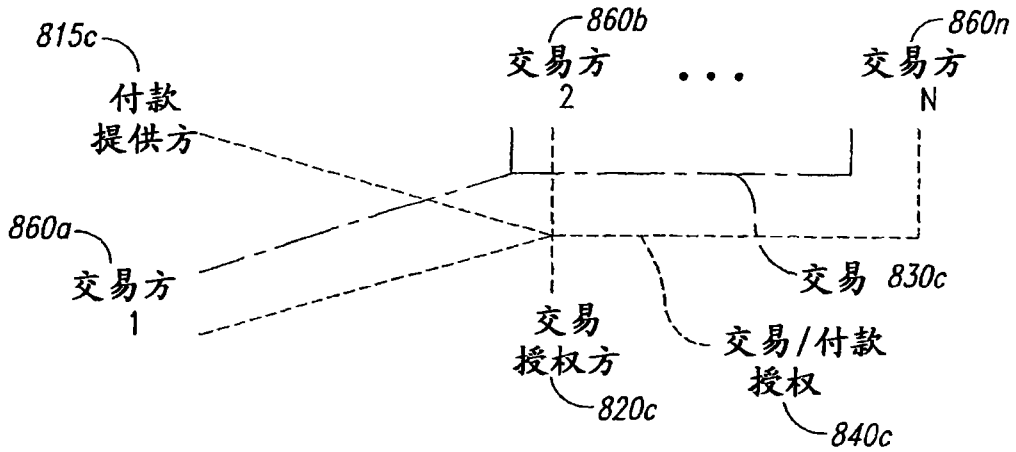


图 8C

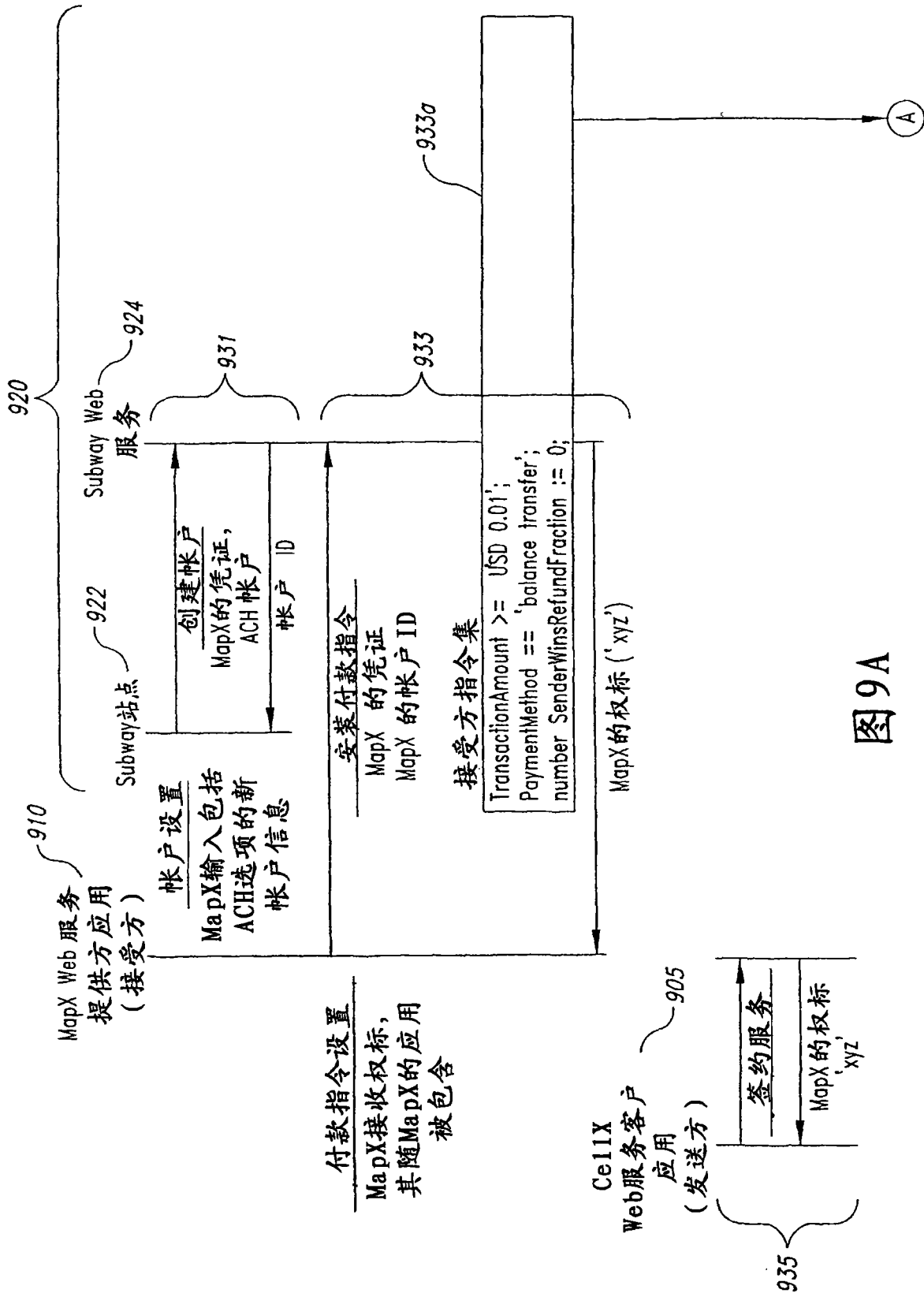


图9A



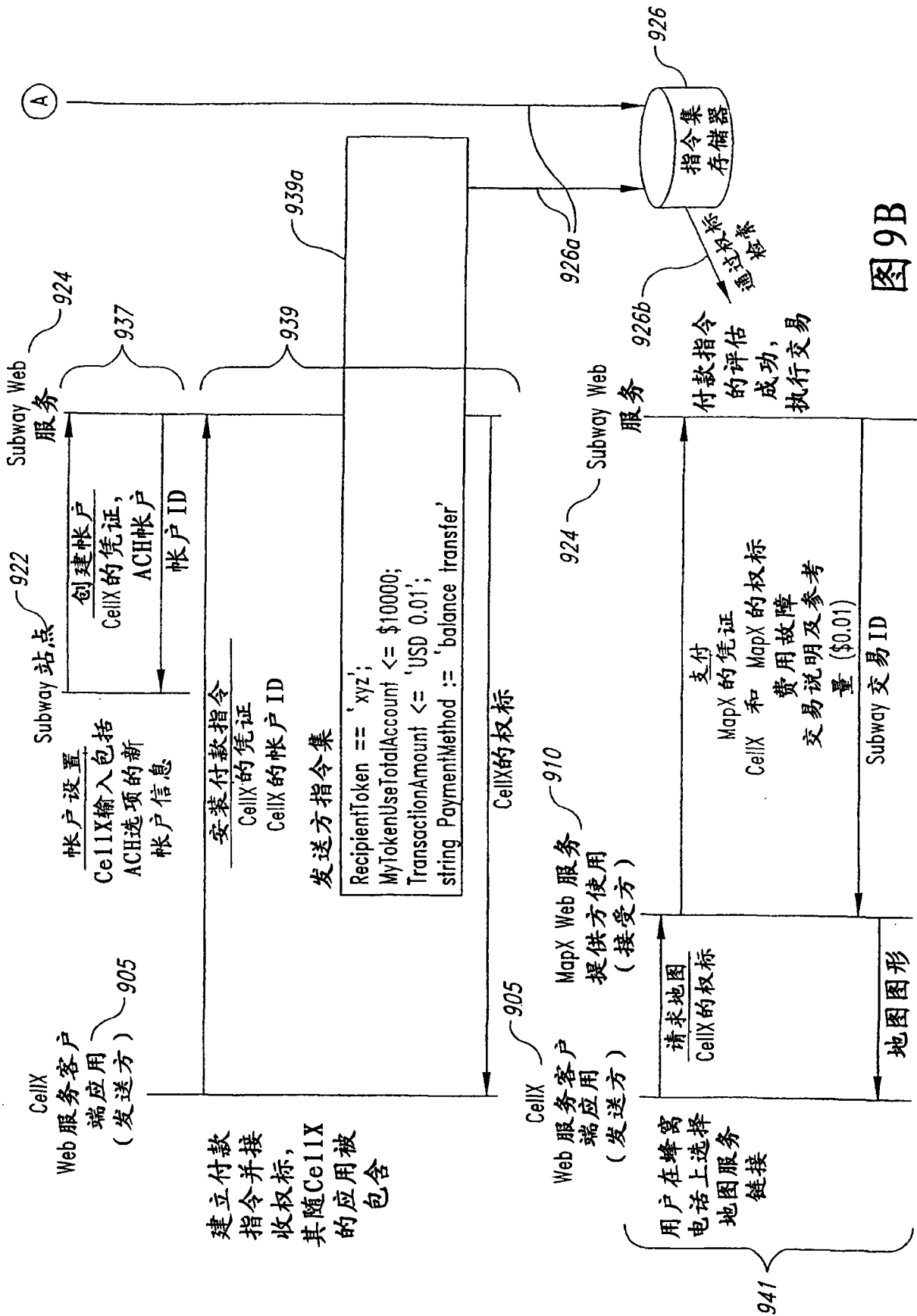


图9B

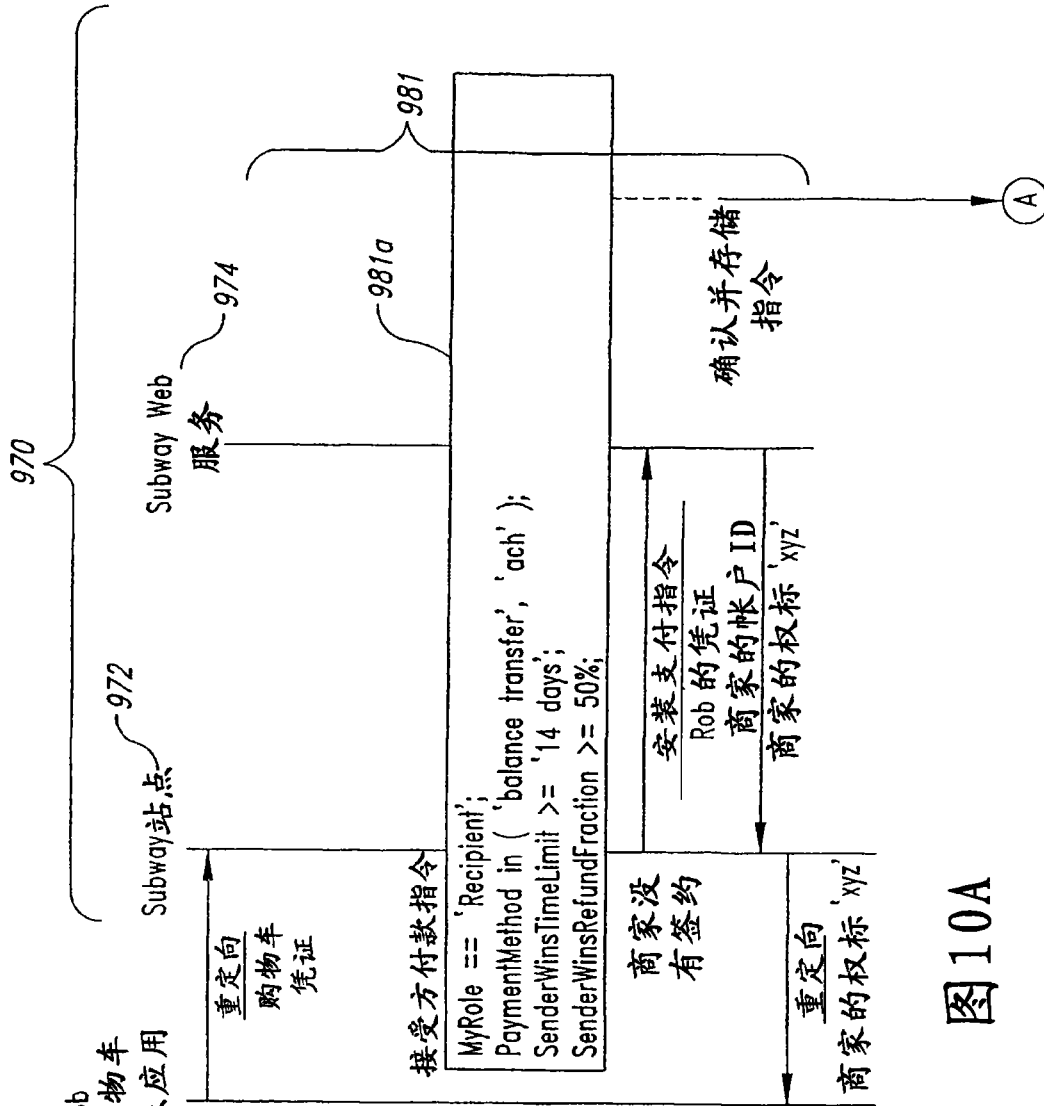


图10A

960 Rob的购物车Web站点应用

商家通过Rob提供的UI建立购物车控制

商家返回“添加购物车”和“进到结帐”URL w/商家权标“xyz”作为参数。商家包括具有其Web站点应用的那些

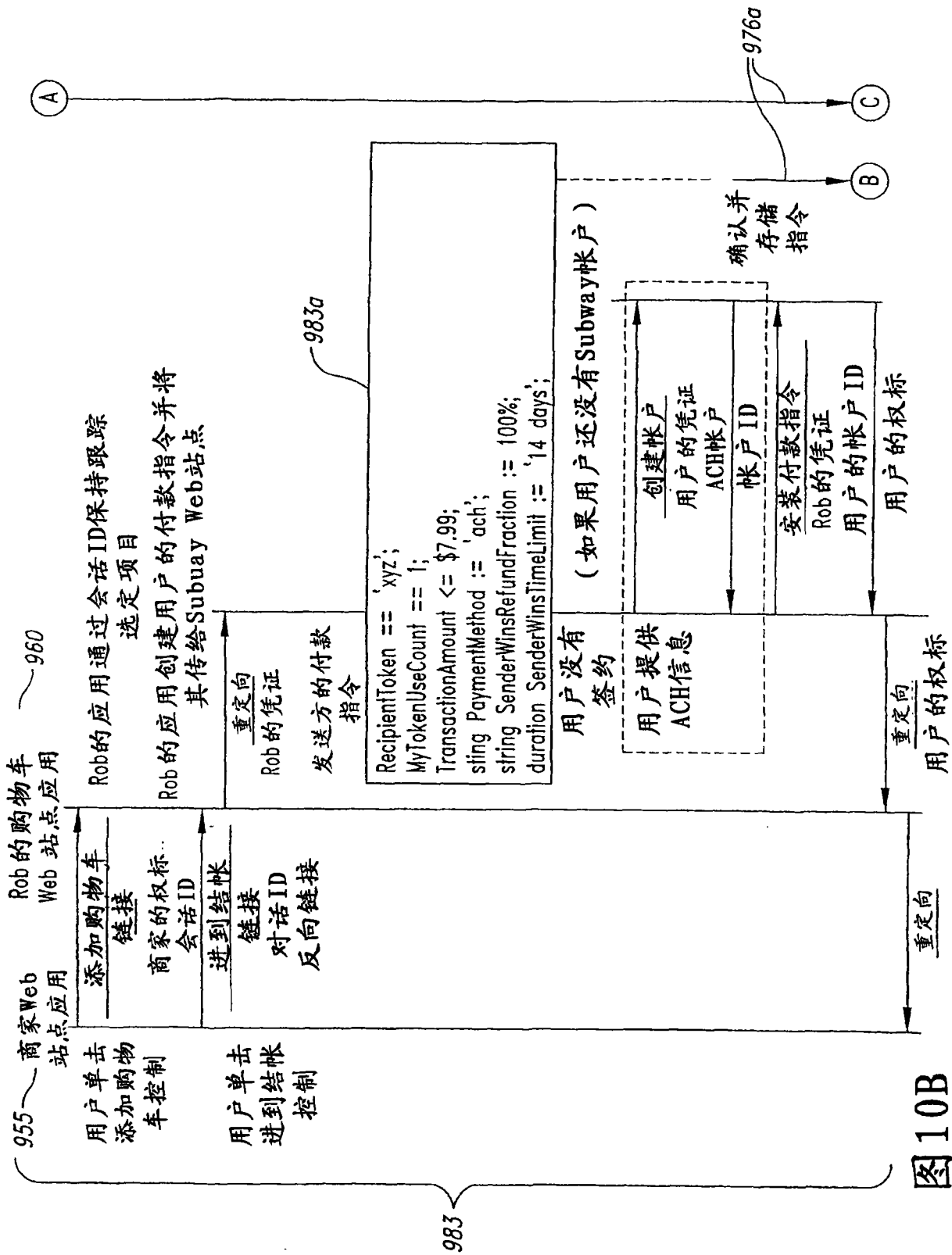


图10B

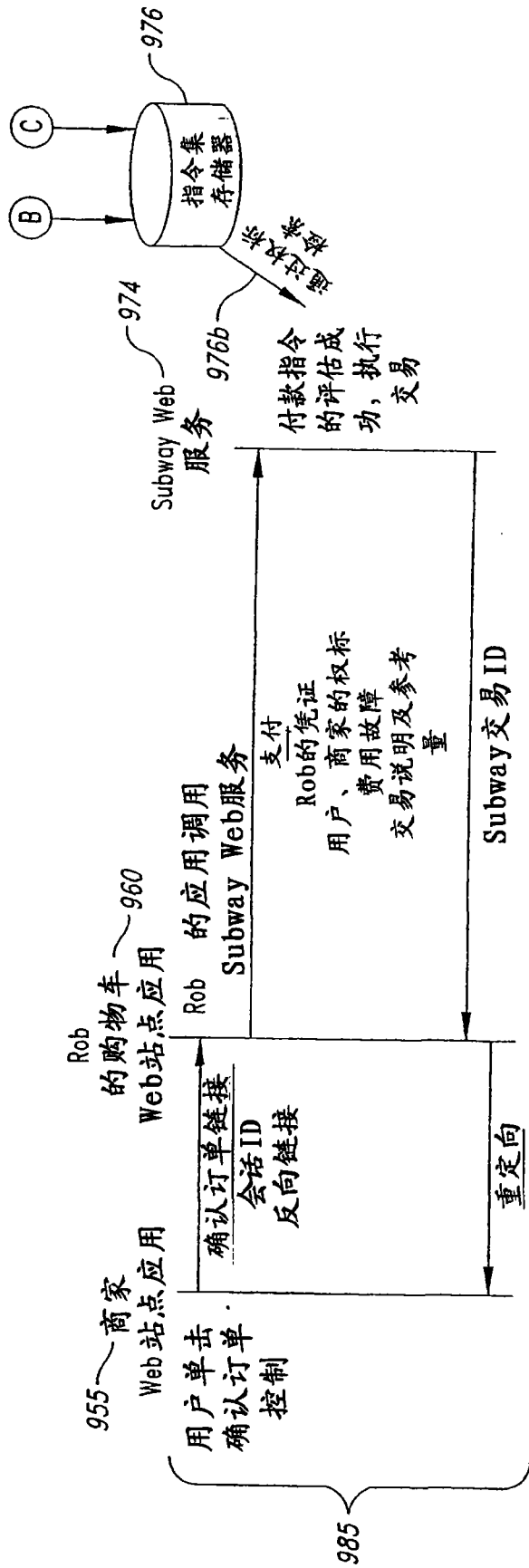


图10C