



[12] 发明专利申请公开说明书

[21] 申请号 200410055737. X

[43] 公开日 2005 年 2 月 9 日

[11] 公开号 CN 1577329A

[22] 申请日 2004. 7. 29

[21] 申请号 200410055737. X

[30] 优先权

[32] 2003. 7. 29 [33] US [31] 10/630,130

[71] 申请人 微软公司

地址 美国华盛顿州

[72] 发明人 D·C·斯蒂尔 B·杜威

R·纳加 V·V·戈特格

[74] 专利代理机构 上海专利商标事务所

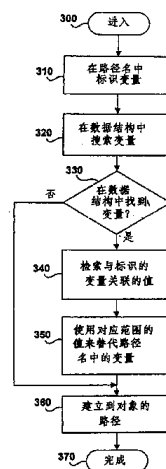
代理人 谢喜堂

权利要求书 2 页 说明书 6 页 附图 4 页

[54] 发明名称 用于后绑定/动态路径名解析的方法和装置

[57] 摘要

可以由在用户或内核模式操作的变量标识器、在内核模式中操作的路径名引擎和数据结构来执行后绑定/动态路径名解析。变量标识器在路径名中标识变量。路径名引擎通过将所述变量转换为字符串来估算所述变量,以建立到对象文件的路径。路径名引擎通过将所述变量映射到储存在所述数据结构中的对应值对所述变量确定潜在字符串值的范围。每一变量具有较佳定义的范围的值,因为唯一可用的值是被选择来储存在所述数据结构中的值。用于建立值的机制是可扩充的,使得在任何时刻可以在数据结构中实现变量/值对。



1. 一种具有计算机可执行组件的计算机可读媒质，其特征在于，它包括：
一数据结构，它包括变量/值映射；
- 5 一路径名解析器，它被配置成在路径名中标识变量、从所述数据结构中检索与所述标识的变量关联的值以及修改所述路径名，使得采用来自所述数据结构的所述值替换所述变量。
 2. 如权利要求1所述的计算机可读媒质，其特征在于，所述路径名解析器包括一变量标识器，它被配置成在所述路径名中标识所述变量。
 - 10 3. 如权利要求1所述的计算机可读媒质，其特征在于，所述路径名解析器包括一路径名引擎，它被配置成通过执行以下步骤来标识所述变量：
在所述数据结构中搜索所述变量；
访问所述对应的值；以及
在所述路径名中插入所述值代替所述变量，使得所述路径标识对象的定位。
 - 15 4. 如权利要求1所述的计算机可读媒质，其特征在于，所述数据结构在操作系统的内核模式中实现。
 5. 如权利要求1所述的计算机可读媒质，其特征在于，所述变量/值对由用户在所述数据结构中定义。
 6. 如权利要求1所述的计算机可读媒质，其特征在于，所述变量/值对由所述
20 计算机可读媒质在其中操作的上下文在所述数据结构中定义。
 7. 一种具有计算机可执行指令的计算机可读媒质，其特征在于，所述计算机可执行指令包括：
接收包括变量的路径名；
通过将所述变量映射到数据结构中的对应值来解析所述路径名；
25 向由所述解析的路径名指向的对象返回一句柄；以及
通过向所述数据结构添加变量/值对来扩展所述数据结构。
 8. 如权利要求7所述的具有计算机可执行指令的计算机可读媒质，其特征在于，所述计算机可执行指令还包括在所述路径名中标识所述变量。
 9. 如权利要求8所述的具有计算机可执行指令的计算机可读媒质，其特征在
30 于，所述计算机可执行指令还包括在操作系统的内核中实现的数据结构中搜索所述

变量。

10. 如权利要求 9 所述的具有计算机可执行指令的计算机可读媒质，其特征在于，所述计算机可执行指令还包括在所述路径中插入对应于所述变量的所述值代替所述变量。

- 5 11. 一种使用可扩充数据结构编码的计算机可读媒质，其特征在于，所述数据结构包括：

第一字段，包括对包括在路径名中的变量的标识；以及

第二字段，包括用于所述变量的值，所述路径名包括可操作来指向对象的值。

12. 如权利要求 11 所述的使用可扩充数据结构来编码的计算机可读媒质，其
10 特征在于，所述数据结构储存在操作系统的内核中。

13. 如权利要求 11 所述的使用可扩充数据结构来编码的计算机可读媒质，其特征在于，所述数据结构可通过添加变量/值对来扩展。

14. 一种计算机实现的方法，其特征在于，它包括：

在由请求访问对象的组件提供的路径名中标识变量；

- 15 将所述变量映射到数据结构中对应的值；

通过用所述对应值在路径名中替换所述变量来修改所述路径名，使得所述解析的路径名创建指向所述对象的路径；以及

将所述修改的路径名返回到所述请求组件。

15. 如权利要求 14 所述的计算机实现的方法，其特征在于，它还包括接收包
20 括变量的路径名。

16. 如权利要求 14 所述的用于解析路径名的计算机实现的方法，其特征在于，它还包括在操作系统的内核中实现的所述数据结构中搜索所述变量。

17. 一种具有用于执行权利要求 14 所述的方法的计算机可执行指令的计算机可读媒质。

用于后绑定/动态路径名解析的方法和装置

5 技术领域

本发明涉及计算系统，尤其涉及标识储存在计算系统上的资源的定位的路径名。

背景技术

10 在计算机操作系统中，路径名是标识文件或目录的定位的字母数字字符的序列。最简单类型的路径名是文件名本身。当由文件名指定路径名时，操作系统在当前工作目录中查找文件。如果文件驻留在不同的目录中，指定路径使操作系统能够定位该文件。将路径名翻译成操作系统语言的过程称为路径名解析。

15 现有系统的一个问题是路径名本质上是静态的。当前，还没有用于在路径用户的上下文中可定制或估算的路径名组件的机制。一旦创建了路径名，不论哪一用户登录、路径名解析出现在哪一机器上等等，该相同路径名都将解析为相同的数据。这一局限已经困扰计算机系统用户和设计者一段时间了。

直到现在，本领域的技术人员仍然不懂用于动态路径名解析的机制。

20 发明内容

依照本发明的一个方面，具有计算机可执行组件的计算机可读媒质包括包含变量/值映射的数据结构和路径名解析器。路径名解析器被配置以在路径名中标识变量、从数据结构中检索与所标识的变量关联的值、并且修改路径名，使得采用来自数据结构的值替换该变量。

25 依照本发明的另一方面，一种计算机可读媒质具有计算机可执行指令。计算机可执行指令包括：接收包括变量的路径名；通过将该变量映射到数据结构中对应的值来解析路径名；向由所解析的路径名指向的对象返回一句柄；以及通过向数据结构添加变量/值对来扩展该数据结构。

依照本发明的再一方面，使用可扩充数据结构对计算机可读媒质进行编码。

30 可扩充数据结构包括第一字段和第二字段。第一字段包括对包括在路径名中的变量

的标识。第二字段包括该变量的值。路径名包括可操作来指向对象的值。

- 依照本发明的再一方面，一种计算机实现的方法包括：在由请求对对象的进行访问的组件所提供的路径名中标识变量；将该变量映射到数据结构中的相应值；通过使用相应值替换路径名中的变量来修改路径名，使解析的路径名创建指向该对象的路径；以及向发出请求的组件返回经修改的路径名。

附图说明

图 1 所示是可以在本发明的一个示例性实施例中使用的示例性计算装置。

图 2 所示是用于实践本发明的示例性环境的结构图。

- 图 3 所示是依照本发明执行的路径名解析过程的示例性概览逻辑流程图。

图 4 所示是依照本发明执行的数据结构扩展过程的示例性概览逻辑流程图。

具体实施方式

- 简言之，本发明涉及用于后绑定 (late-binding) / 动态路径名解析的系统和方法。路径名解析可由变量标识器、路径名引擎和数据结构执行。在请求通过路径名对对象进行访问时，变量标识器在路径名中标识变量。路径名引擎通过参考具有变量/值映射的数据结构估算该变量。数据结构可储存在当前用户的描述表 (context) 中。路径名引擎通过使用来自数据结构的其对应值替换路径名中的变量来修改该路径名，并返回修改的路径名。

- 这里将首先参考可以实现本发明的实施例的说明性计算环境来描述本发明。接着，将描述本发明的一个特定实现的详细示例。还会包括对于特定实现的特定细节的替代实现。可以理解，本发明的具体实施形式不限于这里所描述的实施例。

说明性操作环境

- 参考图 1，用于实现本发明的示例性系统包括计算装置，如计算装置 100。在非常基本的配置中，计算装置 100 通常包括至少一个处理单元 102 和系统存储器 104。根据计算装置的确切配置和类型，系统存储器可以是易失 (如 RAM)、非易失 (如 ROM、闪存等等) 或两者的某一组合。系统存储器 104 通常包括操作系统 105、一个或多个程序模块 106，并可包括程序数据 107。依照本发明，操作系统 105 还可包括路径解析器 120。这一基本配置在图 1 中由虚线 108 内的组件说明。

计算装置 100 可具有另外的特点或功能。例如，计算装置 100 也可包括另外的数据存储设备（可移动和/或不可移动），如，磁盘、光盘或带。这类另外的存储在图 1 中由可移动存储 109 和不可移动存储 110 说明。计算机存储媒质可包括以用于储存信息的任一方法或技术实现的易失和非易失、可移动和不可移动媒质，信息如计算机可读指令、数据结构、程序模块或其它数据。系统存储器 104、可移动存储 109 和不可移动存储 110 都是计算机存储媒质的示例。计算机存储媒质包括但不限于，RAM、ROM、EEPROM、闪存或其它存储技术、CD-ROM、数字多功能盘（DVD）或其它光存储、磁盒、磁带、磁盘存储或其它磁存储设备、或可以用来储存所期望的信息并可由计算装置 100 访问的任一其它媒质。计算装置 100 也具有（多个）输入设备 112，如键盘、鼠标、输入笔、语音输入设备、触摸输入设备等等。也可包括（多个）输出设备，如显示器、扬声器、打印机等等。这些设备在本领域中是众所周知的，并且不需要在这里详细讨论。

计算装置 100 也可包含允许该装置如通过网络与其它计算装置 118 进行通信的通信连接 116。通信连接 116 是通信媒质的一个示例。通信媒质通常可由诸如载波或其它传输机制中的已调制数据信号中的计算机可读指令、数据结构、程序模块或其它数据包含，并包括任一信息传送媒质。术语“已调制数据信号”指以对信号中的信息进行编码的方式对其一个或多个特征进行设置或改变的信号。作为示例而非局限，通信媒质包括有线媒质，如有线网络或直接连线连接、以及无线媒质，如声学、RF、红外和其它无线媒质。这里使用的术语计算机可读媒质包括存储媒质和通信媒质。

路径名解析器的结构

参考图 2，示出了用于实践本发明的示例性环境的结构图。图 2 所示的示例性环境是用于文件系统的路径/链接子系统环境 200，包括依照本发明的路径名解析器 120。该环境包括请求组件 205，可以是请求通过路径名 206 对对象（如，对象 212）的访问的任一组件。例如，请求组件 205 可以是以下的任一个或多个：电子邮件消息中的嵌入式路径或统一资源定位器（URL）；以及具有包括在脚本中的变量的路径。

在本实施例中，用来指对象 212 的路径名 206 包括变量。例如，这一路径名的一个示例可以是“¥¥server¥share¥@user¥phonelist.doc”。在该示例中，“user

(用户)”项通过使用“@”字符被标识为变量。也可以使用其它字符。

路径名解析器 120 包括变量标识器 210、路径名引擎 220 和数据结构 222。在这一特定实现中，变量标识器 210 在用户模式中操作，路径名引擎 220 和数据结构驻留在内核模式中。路径名引擎 220 响应请求组件 205 并向变量标识器 210 请求变量估算。作为替代，请求组件 205 直接与变量标识器 210 进行通信来请求变量估算。

变量标识器 210 在路径名 206 中标识变量。在这一特定实现中，通过在路径名 206 中查找独特前缀来标识变量，在本示例中为“@”字符。独特前缀可以是一个或多个字符的任一组合。变量标识器 210 将标识的变量返回到路径名引擎 220。

路径名引擎 220 通过参考数据结构 222 估算变量。数据结构 222 储存大量的映射到对应值的变量名。这些值可以是字符串，或者可以是导致串的到任意可执行代码的参考点。路径名引擎 220 在数据结构 222 中搜索变量。仅当在数据结构中储存了值时，才找得到对应的值，因此每一变量具有适当定义的值范围，因为唯一可用的值是储存在数据结构 222 中的值。如果该变量在数据结构 222 中不存在，则将该路径名标识为字符串而非包括变量的路径名。这允许将合法的字符用作标识变量的前缀。作为替代，可以使用不合法的字符来标识变量，在这一情况下，如果变量在数据结构 222 中不存在，将返回错误。

如果在数据结构中存在对该变量的正确值，则将该值返回到变量标识器 210。变量标识器 210 然后可以通过用该变量的值替换该变量来修改由请求组件 205 提供的路径名 206。以这一方式，修改的路径名指向对象 212。变量标识器然后可以请求将修改的路径名解析为对对象 212 的句柄并将该句柄返回到请求组件 205。

在一个实施例中，数据结构 222 可以对每一用户唯一，使不同的用户拥有一组不同的变量/值映射。这允许原文相同的路径名（包括变量）基于登录的是哪一用户而映射到不同的对象。例如，如果在动态路径名中使用了诸如“@username”的变量，该变量将基于是哪一用户登录到计算系统而计算为不同的值。在另一示例中，诸如“@L-146”的变量可以指办公室位置。该变量可以在标识该办公室内的打印机的路径名中使用，但是具有一个基于该办公室的当前占有者的名字。因此，数据结构 222 可包括将办公室位置变量映射到该办公室的当前占有者的用户名的表。许多其它的替换方案对本领域的技术人员也是显见的。

在该实现中，任何特许的应用能够向数据结构 222 写入值和/或变量，由此令该机制可扩充。作为替代，数据结构 222 可以是大小固定的，使用有限数量的值，

使得无法任意添加值。可以在具有由文件系统 200 的需求所确定的一个潜在值范围的数据结构 222 中定义独特的变量。

- 可以以许多方式定义对应于变量的值。例如，值可以是用户定义的或者值可以是上下文特定的（如，可以由操作系统或环境的状态定义值）。可以提供广泛使用的一些基本值，如对应于定义为“user（用户）”的变量的值。

后绑定/动态路径名解析的过程

- 参考图 3，示出了依照本发明的路径名解析过程的示例性概览逻辑流程图。过程在起始块 300 进入，其中，请求组件请求对由路径名标识的对象的访问。过程继续到块 310。

在块 310，在路径名中定义变量。可以通过将路径名分析成其构成部分并搜索独特的前缀字符，例如“@”来标识变量。如果找到变量，过程继续到块 320。

- 在块 320，在数据结构，如具有变量/值映射的表中执行对变量的搜索。在判别块 330，确定是否在数据结构中找到用于该变量的条目。如果在数据结构中找到该变量，则过程继续到块 340。如果数据结构中未找到该变量，则过程继续到块 360，其中，将该变量作为不需要解析的字符串处理。

- 在块 340，从数据结构中检索与该变量关联的值，过程继续到块 350。在块 350，使用对应的值来替换路径名中的变量来组成完整的路径名，过程继续到块 360。在块 360，将完整的路径名返回到请求组件。过程继续到块 370，完成了路径名解析。

向数据结构添加变量/值对的过程

参考图 4，示出了依照本发明执行的用于扩展数据结构的示例性过程的逻辑流程图。可以通过添加新的变量/值对来扩展该数据结构。

- 过程在块 400 开始，其中，系统管理员决定向数据结构放入更多的变量和/或对应值。在判别块 410，确定该操作系统是否支持后绑定/动态路径名。如果该操作系统支持后绑定/动态路径名，则过程继续到块 430。如果该操作系统不支持后绑定/动态路径名，则过程继续到块 420。在块 420，发布错误消息，通知系统管理员操作系统不支持后绑定/动态路径名，过程继续到块 460。

- 在块 430，向系统管理员提示输入变量/值对。过程继续到块 440。在块 440，系统管理员向系统输入变量/值对。过程继续到块 450。在块 450，向数据结构添加

变量/值对。过程在块 460 结束。

- 可以理解，依照本发明向数据结构添加变量/值对也可以由脚本执行，如在用户登录时。此外，可以设置包括预定义变量/值映射的政策。类似地，具有足够的权限来以保护模式改变数据结构的软件组件可以直接访问并修改数据结构中的变量/值对。以这一方式，可以创建在特定用户、若干组件在其上执行的特定硬件或软件的上下文中、或基于一些其它标准解析的变量/值映射。
- 5

以上说明、示例和数据提供了本发明的制造和组成部分的使用的完整描述。由于可以在不脱离本发明的精神和范围的情况下作出本发明的许多实施例，因此本发明包含在所附权利要求书中。

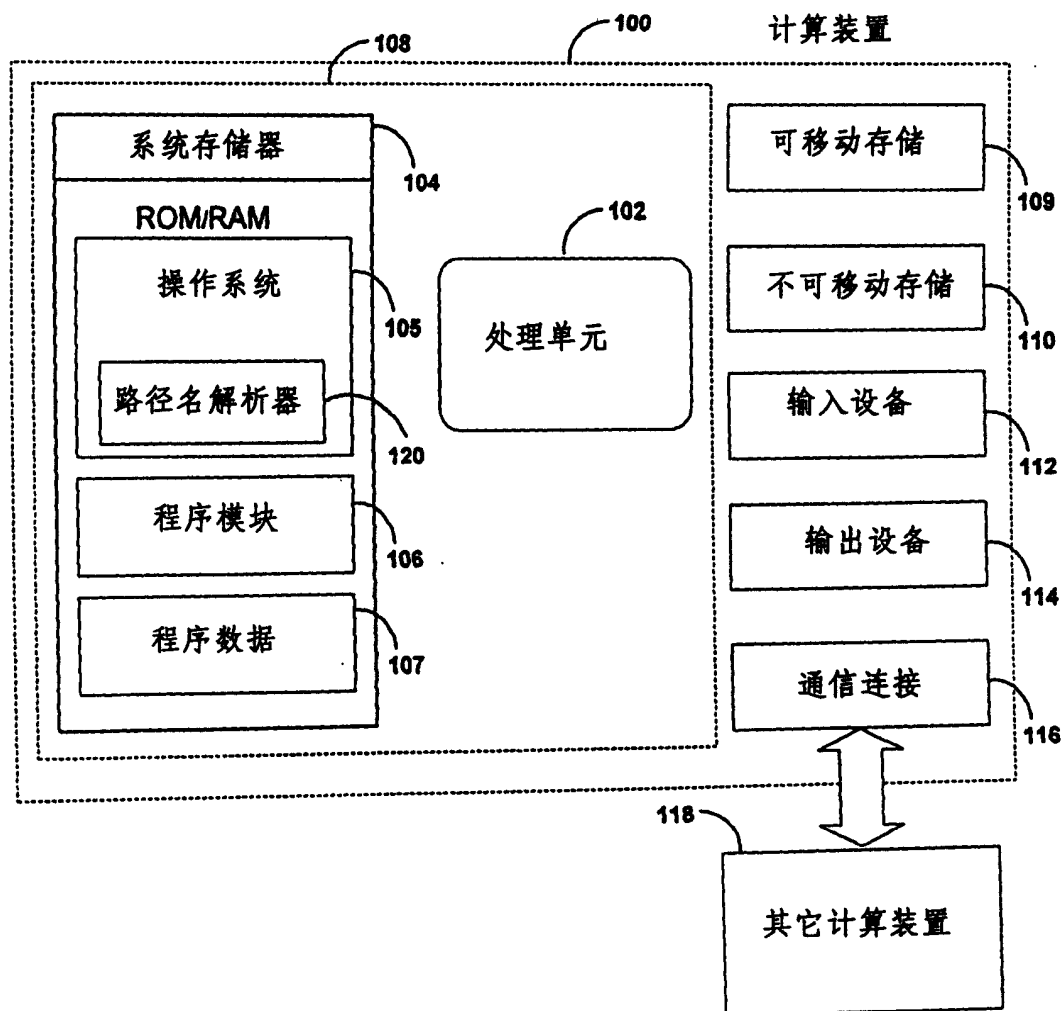


图 1

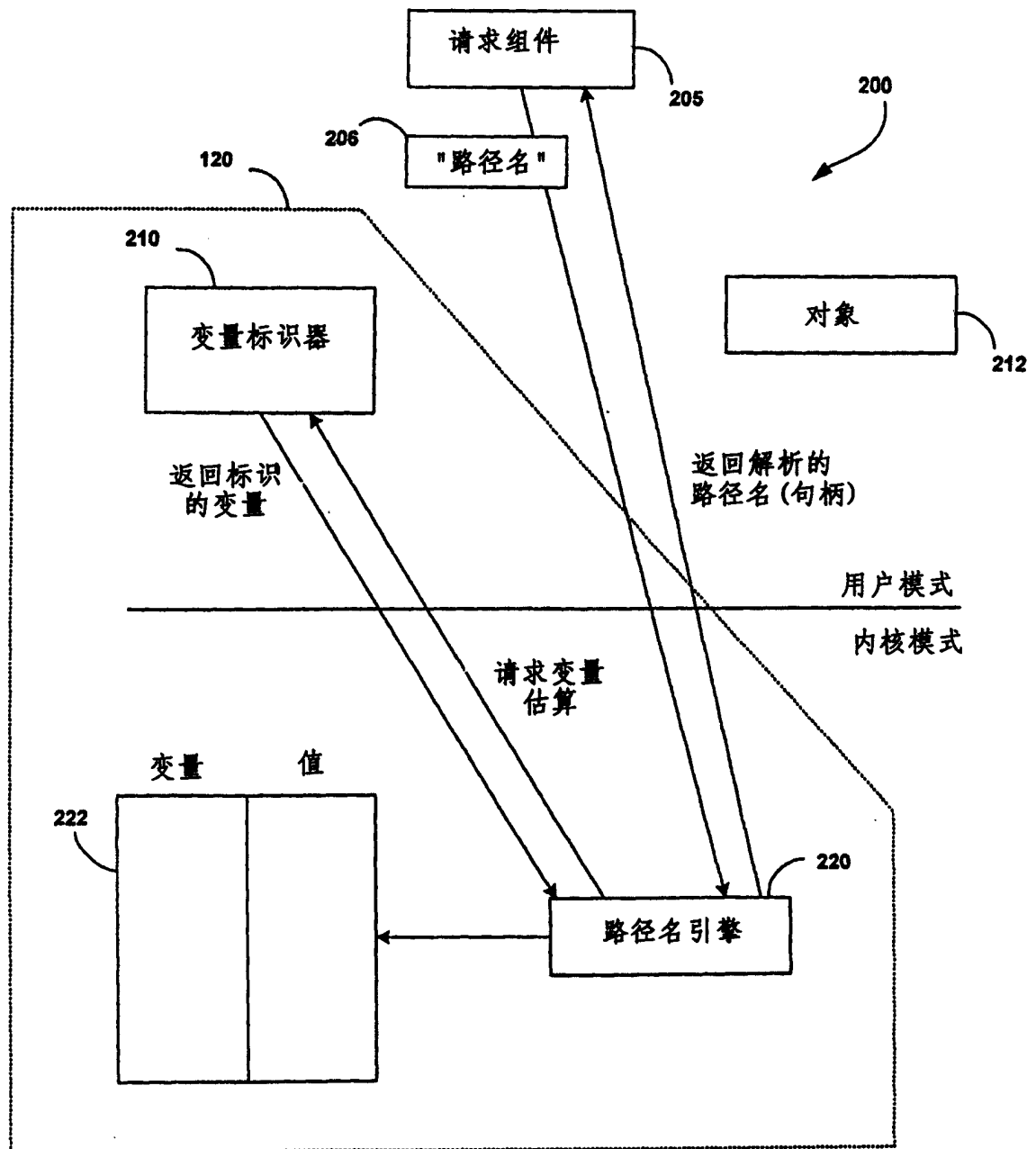


图 2

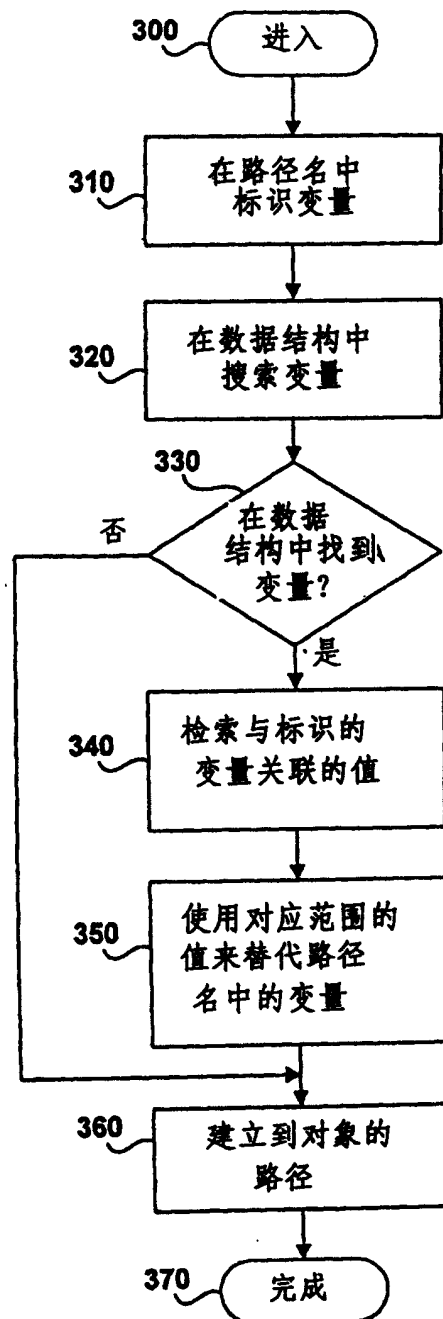


图 3

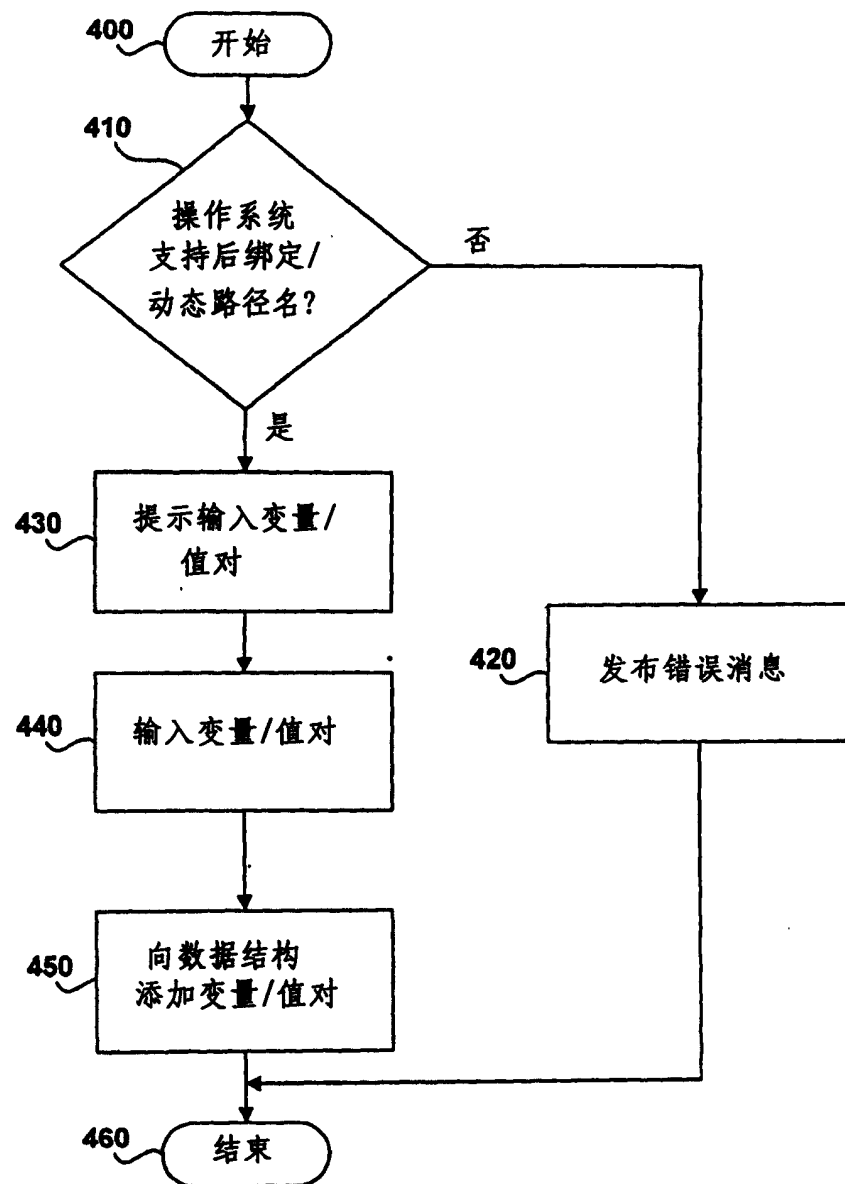


图 4