[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
G06F 9/44 (2006.01)

[12] 发明专利说明书

专利号 ZL 200410043491.4

[45] 授权公告日 2009 年 2 月 25 日

[11] 授权公告号 CN 100464294C

[22] 申请日 2004.5.12
[21] 申请号 200410043491.4
[30] 优先权
[32] 2003.5.12 [33] US [31] 10/438，234
[73] 专利权人 微软公司
地址 美国华盛顿州
[72] 发明人 J·P·斯诺弗
J·W·特鲁尔三世 D·W·雷
K·普西帕瓦南姆
[56] 参考文献
US6405365 B1 2002.6.11
US6286035 B1 2001.9.4
US6266666 B1 2001.7.24
CN1402559 A 2003.3.12
审查员 王 越

[74] 专利代理机构 上海专利商标事务所有限公司
代理人 陈 斌

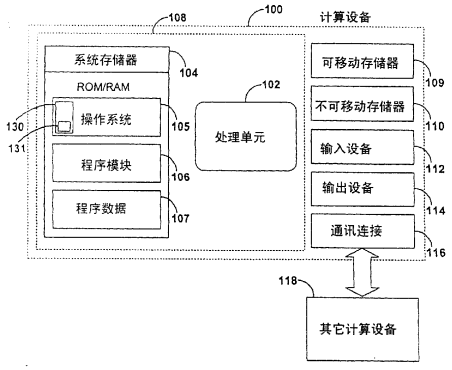
权利要求书 2 页 说明书 15 页 附图 4 页

[54] 发明名称

对命令的输入参数执行基于反射的处理的方法和系统

[57] 摘要

本发明的目标是基于反射的外壳，它提供对一命令的输入参数的基于反射的处理。 基于反射的处理包括语法分析，数据生成，数据验证，对象编码，对象处理，文档等。 基于反射的反壳提供一机制，用于使用类对输入参数规定方法。 该方法包括接收可语法分析的流，它包括一命令及至少一个参数。 检索描述对该命令的期望的参数的定义信息。使用定义信息创建一对象，用于以按该期望的参数的描述的格式存储至少一个参数。



1、一种在输入到一命令的参数上完成基于反射的处理的方法，包括：

接收包括与所述命令相关的标识的可语法分析的流；

基于所述标识检索描述用于所述命令的期望的参数的定义信息；

基于所述定义信息创建一对象；

按照与所述期望的参数相关联的定义信息在所述对象中存储从所述可语法分析的流中获得的参数；

在所述参数上运用处理控制指令，以在将带有所述参数的对象提供给所述命令之前操作所述参数，所述处理控制指令与包括可被处理的字符串和集合的特定大小限制的定义信息相关联；

应用文档控制指令到所述参数，所述文档控制指令在被请求时产生有关所述参数的文字信息，所述文档控制指令与所述定义信息相关联，并在遇到无效语法时提供正确语法的描述；以及

将所述对象提供给所述命令，所述对象具有由所述命令可调用的方法。

2、如权利要求1的方法，其特征在于，在命令行上输入所述可语法分析的流。

3、如权利要求1的方法，其特征在于，经语音输入产生所述可语法分析的流。

4、如权利要求1的方法，其特征在于，通过用户界面或脚本产生所述可语法分析的流。

5、如权利要求1的方法，其特征在于，检索定义信息包括识别与所述命令相关的类。

6、如权利要求5的方法，其特征在于，检索定义信息还包括从所述类读出元数据。

7、如权利要求1的方法，其特征在于，还包括当正产生可语法分析的流时提供智能感觉，以便根据该定义信息自动完成可语法分析的流。

8、如权利要求1的方法，其特征在于，还包括应用将所述参数与所述期望的参数关联的语法分析控制指令，所述语法分析控制指令与所述定义信息相关联。

9、如权利要求1的方法，其特征在于，还包括应用确定获得所述参数的方式的语法分析控制指令，所述语法分析控制指令与所述定义信息相关联。

10、 如权利要求1的方法，其特征在于，还包括应用数据验证控制指令，以

确定所述可语法分析的流是否满足由与所述定义信息相关联的所述数据验证控制指令规定的准则。

11、 如权利要求1的方法，其特征在于，还包括在所述参数上应用数据生成控制指令，以产生存储在所述对象中的一组信息，所述数据生成控制指令与所述定义信息相关联。

12、 如权利要求1的方法，其特征在于，还包括在所述参数上应用第二处理控制指令，以在将带有所述参数的对象提供给所述命令之前操纵所述参数，所述处理控制指令与包括对象要被编码的特定类型的定义信息相关联。

13、 一种在输入到一命令的参数上完成基于反射的处理的系统，该系统包括：

接收包括与所述命令相关联的标识的可语法分析的流的装置；

基于所述标识检索描述用于所述命令的期望的参数的定义信息的装置；

基于所述定义信息创建一对象的装置；

按照与所述期望的参数相关联的定义信息在所述对象中存储从所述可语法分析的流中获得的参数的装置；

在所述参数上运用处理控制指令，以在将带有所述参数的对象提供给所述命令之前操作所述参数的装置，所述处理控制指令与包括可被处理的字符串和集合的特定大小限制的定义信息相关联；

应用文档控制指令到所述参数的装置，所述文档控制指令在被请求时产生有关所述参数的文字信息，所述文档控制指令与所述定义信息相关联，并在遇到无效语法时提供正确语法的描述；以及

将所述对象提供给所述命令的装置，所述对象具有由所述命令可调用的方法。

14、 如权利要求13的系统，其特征在于，所述可语法分析的流是在一命令行上输入的。

15、 如权利要求13的系统，其特征在于，所述基于所述标识检索描述用于所述命令的期望的参数的定义信息的装置还包括用于识别与所述命令相关的类的装置。

16、 如权利要求15的系统，其特征在于，所述基于所述标识检索描述用于所述命令的期望的参数的定义信息的装置还包括从所述类读出元数据的装置。

对命令的输入参数执行基于反射的处理的方法和系统

技术领域

本发明涉及对命令的输入参数执行基于反射的处理的方法和系统。

背景技术

多用户计算机系统的系统管理是十分专门化的。负责系统管理的系统管理员希望知道及理解使用不一致的语法的诸命令，错误报告等。因为命令是由若干不同的软件开发者使用自己的风格写的，就发生命令之间的这些不一致性。这些不一致性对系统管理员造成了困难。

一种困难涉及到命令的可用性。存在不一致的命令使得系统管理员更难以掌握及使用。例如，某些开发者在参数之间喜欢使用划（“-”），另外一些喜欢正斜杠（“/”），又一些喜欢使用其他独特的语法。各开发者在报告错误消息时也各具有他们自己的风格。因此，系统管理员必须知道每条命令的语法，并掌握每个错误消息的格式。

另外的困难关系到命令的维护。例如，当写一条命令的开发者离开时，其他开发者必须查看代码本身或查看关于该命令的文档。所有这些方法均是非常不希望的。

在试图使命令更一致之前着眼于提供完成常用功能的例行程序库。虽然此库能减少用于完成由库例行程序提供的常用功能的代码的量，开发者们仍使用他们自己的风格处理由使用任何的库例行程序引起的错误条件。此外，库的使用不影响每条命令需要包含从命令行获得输入参数的逻辑。因此，虽然使用例行程序库能减少需要写的代码的量，对每条命令仍然有相当数量的复制代码要产生，以便完成语法分析，数据验证，和错误报告。

因此需要一个环境，其中以更一致的方式获取和处理对命令的输入参数，同时减少对该命令所需的代码的量，并在对所有命令的参数获取阶段中提供一致的错误报告。

发明内容

本发明的目标是基于反射的外壳(reflection-based shell)，它对命令提供输入参数的基于反射的处理。基于反射的处理包括语法分析，数据生成，数据验证，对象编码，对象处理，文档等。基于反射的处壳提供一机制，用于使用类来规定对输入参数的文法。第三方开发者使用类规定对他们的命令的文法。在操作中，本发明接收可语法分析的流。可语法分析的流能从命令行，语音输入，脚本等获得。可语法分析的流包括命令和至少一个参数。检查描述对命令的预期参数的基于语法分析的流，定义信息。使用定义信息创建一对象（即由开发者创建的类的实例）。该对象以按预期的参数的描述的格式存储至少一个参数。然后该对象转到作单独处理的命令。定义信息能包括控制指令(directive)，它规定在可语法分析的流上拟完成的活动，例如如何映射参数到预期的参数，如何获得参数（如交互式的）等。控制指令也能包括关于语法分析，验证，文档，数据生成，和数据处理的活动。

因此，本发明的一个优点在于命令的开发者对他们的命令的输入参数很容易地规定文法，而不需要写分析命令行以得到输入参数，或验证输入参数的逻辑。这样，本发明就减少了开发者需要写的代码的量，并使得对命令的语法更加一致，而十分通用。

附图说明

图1示出能用于本发明的一个示例性实施例的示例性计算设备。

图2是功能流图，示出在通过按本发明的基于反射的外壳中的语法分析程序和引擎对命令行的处理。

图3是数据结构的实施例，用于按本发明规定对命令的输入参数的文法。

图4是逻辑流程图，示出在本发明的基于反射的外壳中处理在命令行输入的输入参数的示例性过程。

具体实施方式

简言之，本发明的目标是基于反射的外壳，它提供对命令的输入参数的基于反射的处理。阅读下面的详述后将明白，本发明使第三方开发者需要写的代码的量最少，并使系统管理员为完成系统管理的任务需要知道的知识最少。从而，本发明大大减轻了系统管理员的任务。此外，本发明对输入参数提供更一致的语法，并为与输入参数有关的处理提供常用的功能。

图1示出能在本发明的示例性实施例中使用的示例性计算设备。在十分基本的配置中，计算设备100通常至少包括一个处理单元102和系统存储器104。根据确切的配置和计算设备的类型，系统存储器104能是易失的（如RAM），非易失的（如ROM，闪存等）或两者的某种组合。系统存储器104通常包含操作系统105，一个或多个程序模块106，且能包括程序数据107。操作系统105包括执行操作系统命令的命令处理器130。命令处理器130包括接收操作系统命令的外壳131（即命令处理界面）。该外壳能显示命令提示，能显示图象用户界面，或用于输入和解释用户输入的其他装置。外壳131验证，输入的命令是否有效，并发送验证的命令到命令处理器130的另外部分来执行。基本配置在图1中由虚线框108由的那些组件示出。

计算设备100能具有另外的特征或功能。例如，计算设备100也能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，或磁带。在图1中那样的另外存储器由可移动存储器109和不可移动存储器110示出。计算机存储介质能包括以任何方法或技术实现的易失和非易失，可移动和不可移动介质，用于存储如计算机可读指令，数据结构，程序模块或其他数据那样的信息。系统存储器104，可移动存储器109和不可移动存储器110均是计算机存储介质的例子。计算机存储介质包括RAM，ROM，EEPROM，闪存或其他存储技术，CD—ROM，数字多功能盘（DVD）或其他光存储器，盒式磁带，磁带，磁盘存储器或其他磁存储设备，或能用于存储希望的信息并能由计算设备100访问的任何其他介质，但不限于这些。任何那样的计算机存储介质能是设备100的一部分。计算设备100也能具有如键盘，鼠标，输入笔，语音输入设备，接触输入设备等的输入设备112。还能包括如显示器，扬声器，打印机等那样的输出设备114。这些设备在业内是众知的，不必在此详细讨论。

计算设备100还能包含通讯连结116，使该设备能如通过网络与其他计算设备118能讯。通讯连结116是通讯介质的一个例子，通讯介质通常由计算机可读指令，数据结构，程序模块或以如载波或其他机制的调制数据信号形式的其他数据实现，并包括信息提交介质。术语“调制数据信号”指的是具有以如在信号中编码信息的方式设置或改变的一个或多个特征的信号。例如，通讯介质包括如有线网络或直线连结的有线介质，和如声音，RF，红外及其他无线介质的无线介质，且不限于这些。这里使用的术语“计算机可读介质”包括存储介质和通讯介质两者。

图2是功能流图，示出在通过按本发明的基于反射的外壳的语法分析器202和引擎204的命令行250的处理。示例的命令250将若干命令（即处理(process)命令260，where命令262，分类(sort)命令264，和table命令266）流水线化。然而，下面 讨论集中在对一个命令（如where命令262）的输入参数的基于反射的处理。对其他命令的基于反射的处理以类似方式完成。命令行250能传送输入参数到任何命令（如，“handlecount>400”被传送到where命令262）。人们注意到，处理命令260不具有任何有关的输入参数。过去，每条命令负责语法分析与该命令相关的输入参数，判断那些输入参数是否有效，且若输入参数无效发出错误消息。因为命令通常由不同的程序员写出，对于在命令行上输入参数的语法不是非常一致。此外，若发生错误，即使对同一错误而言，在各命令之间错误消息也不很一致。

例如，在Unix环境中，“ls”命令和“ps”命令之间具有许多不一致处。虽然两者均接收选项“-w”，由“ls”命令使用的“-w”选项表示页的宽度，而由“ps”命令使用的“-w”选项表示打印宽度输出（实质上不管页的宽度）。与“ls”及“ps”命令相关的求助页也具有若干不一致处，使得在一处用黑体字而在另一处不用，在一处按字母表排列而去另一处不是，在一处需要划线而在另一处不用。

如下面详细讨论，本发明提供更一致的方法，并使每个开发者必须写的复制编码的量最少。基于反射的外壳200提供一语法（如文法），对应的语义（如词典），和使开发者便于得益于由基于反射的外壳200提供的常用功能的参考模型。

在进一步描述本发明之前，提供对出现在本专利说明中的术语的定义。“命令-command”指的是独立的可执行程序。“commandlet-小命令”或“cmdlet”指的是比命令小得多的程序。在一个实施例中，每个cmdlet定义名词-动词对（如在命令行250中的get/process）。当参考按本发明写的命令时，下面讨论使用术语cmdlet。然而，在某些情况，使用更常用术语“command”谈及cmdlet。输入参数指的是对cmdlet的输入字段。变量指的是传送到命令或cmdlet的输入参数，它是argv数组中的单个字符串的等价物，或作为在Request Object中的单个元素传送。如下面将描述，Request Object指的是用于对cmdlet规定文法的机制。变量是选项，选项变量，或在命令名后的操作数中的一个。根据下述命令行给出变量的例子：

findstr/i/d:\winnt;\winnt\system32 aa*b*.ini。

在上述命令行中，“findstr”是变量0，“/i”是变量1，“/d:\winnt;\winnt\system32 ”是变量2，aa*b是是变量3，而“*.ini”是变量4。“选项”是对命令或cmdlet的变量，通常用于规定对程序的默认行为的改变。继续上述的例子命令行，“/i”和“/d”是选项。“选项变量”是跟随某些选项的输入参数。在某些情况，选项变量作为选项包括在同一变量串之中。在另外情况，选项变量作为下一变量包括其中。再次参考上述命令行，“winnet;\winnet\system32”是选项变量。“操作数”是对命令或cmdlet的变量，它通常用作为向程序提供为完成程序处理所必须的信息的对象。操作数在命令行中通常跟在选项之后。再次参考上面的例子命令行，“aa*b”和“*.ini”是操作数。“可语法分析的流”包括变量。

“类成员-class members”指的是元素，如子类，字段，常数，方法，结构，特性，数组，索引，界面，事件，异常等。“控制指令-directive”指的是元数据属性。“类别-category”指的是一组特定类型的控制指令。下面将详细解释，本发明的基于反射的外壳提供若干类别的控制指令类别，如语法分析控制指令，数据生成控制指令等。在每个类别中，基于反射的外壳提供若干控制指令。类别和控制指令能由软件开发者扩展。

参考图2，语法分析程序202将可语法分析的流（如命令行250）分析成Request Object220-226（如where请求222）。每个Request Object220-226与一个cmdlet260-266相关联。简言之，下面结合图3详细描述，Request Object220-226对开发者提供为对cmdlet的输入参数规定文法的方法或机制。Request Object被传送到对应的cmdlet可执行码（如where可执行码232）。然而，下面将结合图4描述，语法分析程序202和引擎204在将Request Object传送到cmdlet可执行码之前，在命令行200上规定的输入参数上完成各种处理。处理包括语法分析，参数验证，数据生成，参数处理，参数编码，和参数文档化。因为语法分析程序202和引擎204在命令行的输入参数上完成常用功能，基于反射的外壳200能对用户产生一致的错误消息。人们认识到，按本发明写的可执行cmdlet230-236比以前系统中的命令需要较少的代码。每个可执行的cmdlet230-236接收对应的Request Object220-226。此外，每个可执行的cmdlet230-236输出被输入到下一管线的cmdlet的对象。通常，这些对象通过将一参考（如句柄）传送到该对象而被输入。然后，可执行cmdlet230-236能在被传送入的对象上完成附加的处理。

图3是用于对cmdlet的输入参数规定文法的数据结构300。本质上，数据结构300提供用于清楚地表达在基于反射的外壳和cmdlet之间的约定的方法。下面

讨论使用由Redmend，WA的微软公司创建的，NET架构描述本发明。然而，能使用其他环境而不偏离本发明的范围。

软件开发者在对应的可执行cmdlet的代码中编码数据结构300。实施此请求的方法和特性确定了，什么输入参数通过命令行向用户揭示。数据结构300是从Request Object类304导出的公共类。软件开发者为数据结构300提供类名302。类名302识别在cmdlet命令行上规定的变量的名。每个命令名302表示动词/名词对。如在图2中示出的示例命令行200中的"get/process"和"format/table"。在如"where"命令那样的命令名中，动词或名词能是隐含的。在一个实施例中人们注意到，类名不等同于cmdlet。在此实施例中，使用其他标识识别cmdlet的名。数据结构300至少包括一个公共成员（如Name 330）。公共成员330，332代表与cmdlet相关的输入参数。每个公共成员330，332能具有一个或多个在每个下列类别中的控制指令：语法分析控制指令310，数据验证控制指令312，数据生成控制指令314，处理控制令316，编码控制指令318，和文档控制指令320。这些控制指令用方括号括起来并描述跟随其后的输入参数。某些控制指令也能在类层次上应用，如用户交互类型的控制指令。数据结构300也能包括专用成员340，语法分析器不将其识别成输入参数。专用成员340能用于存储根据一个控制指令生成的数据。

公共成员的名能用在命令行上以限定命令行上的输入参数的有效性。不然，公共成员能用于根据其在命令行的位置存储输入参数。下面是说明此概念的例子，Request Object是如下：

```
public class FileCopy Request: Request Object
{
[Parsing Parameter Position Attribute (O)]
Public string From;

[Parsing Parameter Position Attribute(1)}
Public string To;
...
}。
```

Parsing Parameter Position Attribute是语法分析控制指令，它描述如何根据位置映射非限定的参数。上面提到，非限定的参数是不使用与输入参数相关的

公共成员名的参数。下面是在“To”和“From成员上应用上面语法分析控制命令之后的合适的语法：

$copy/File-From:a-To:b

$copy/File a b

$copy/File-From:a b

$copy/File a-To:b

$copy/File-To:b-From:a

下面的语法是无效的：

$copy/File-To: b a

$copy/File b-From:a

下面将描述，其他控制指令影响在命令行上规定的输入参数的处理。因此，通过使用控制指令，使cmdlet的开发者能容易地规定对他们的cmdlet的输入参数的文法，并完成在输入参数上的处理而不需要他们产生任何底层逻辑。

控制指令被存储在与cmdlet相关的元数据中。后面结合图4描述，元数据的处理分配在整个基于反射的外壳。例如，可应用性控制指令，文档控制指令，和语法分析导引控制指令在语法分析器的很早阶段处理。一旦语法分析程序结束了所有输入参数的分析，在引擎中处理数据生成控制指令和验证控制指令。

下列表格示出对各种类别的代表性控制指令，以及由基于反射的外壳响应该控制指令完成的处理的说明。

名字	描述
Prerequisite Machine Role Attribute	通知外壳，是否该元素只用于某些机器角色（如文件服务器电子邮件服务器）。
Prerequisite User Role Attribute	通知外壳是否该元素只用于某些用户角色（如域管理员，备份操作员）。

表 1、可应用性控制指令

名字	描述

Parsing Parameter Position Attribute	根据位置映射非限定性参数。
Parsing Variable Length Parameter List Attribute	映射不具有 Parsing Parameter Position Attribute 的参数。
Parsing Disallow Interaction Attribute	当参数的数目小于需的数目时规定活动。
Parsing Require Interaction Attribute	规定，参数通过交互获得。
Parsing Hidden Element Attribute	使参数最终用户不可见。
Parsing Mandatory Parameter Attribute	规定需要该参数。
Parsing Password Parameter Attribute	需要专门处理参数。
Parsing Prompt String Attribute	规定对该参数的提示。
Parsing Default Answer Attribute	规定对参数的默认回答。
Parsing Default Answer Script Attribute	规定得到对参数的默认回答的活动。
Parsing Default Value Attribute	规定对参数的默认值。
Parsing Default Value Script Attribute	规定得到对参数的默认值的活动。

表 2、语法分析导引控制指令

名字	描述
Document Name Attribute	提供参考元素的名字用于交互或求助。
Document Short Description Attribute	提供元素的简单描述。
Document Long Description	提供元素的详细描述。

Attribute	
Document Example Attribute	提供元素的例子。
Document See Also Attribute	提供有关元素的列表。
Document Synopsis Attribute	提供对元素的文档信息。

表 3、文档控制指令

名字	描述
Validation Range Attribute	规定参数必须在某范围内。
Validation Set Attribute	规定参数必须在某集合内。
Validation Pattern Attribute	规定参数必须符合某格式。
Validation Length Attribute	规定字符串必须在长度范围内。
Validation Type Attribute	规定参数必须是某类型。
Validation Count Attribute	规定输入项必须是某数目。
Validation File Attribute	规定对文件的某些特性。
Validation File Attributes Attribute	规定文件必须在规定的范围内。
Validation FileSize Attribute	规定文件必须在规定的范围内。
Validation Network Attribute	规定给定的网络实体支持某些特性。
Validation Script Attribute	规定在使用元素前估计的条件。
Validation Method Attribute	规定在使用元素前估算的条件。

表 4、数据验证控制指令

名字	描述
Processing Trim String Attribute	规定字符串的长度限止。

Processing Trim Collection Attribute	规定集合的大小限止。
Encoding Type Coercion Attribute	规定拟编码的对象的类型。

表5、处理和编码控制命令

在使用.NET架构的实施例中，每个类别具有从基本类别的类（如CmdAttribute）导出的基本类。基本类别类从system.Attribute类导出。每个类别具有预定义的函数（如attrib.func()），它由语法分析程序在类别处理期间调用。cmdlet开发者能创建一客户类别，它从客户类别类（如CmdCustom Attribute）导出。cmdlet开发者还能通过从对现有类别的基本类别类导出控制指令类而扩展现有的类别类，并用它们的实施覆盖预定的函数。cmdlet开发者还能复盖控制指令并增加控制指令到预定的控制指令组。

处理这些控制指令的次序能存储在由语法分析程序可访问的外部数据存储。基于反射的外壳寻找登录的类别并对该类别中的每个控制指令调用一函数（如Process Custom Directive)。因此，通过存储类别执行信息到永久存储中，类别的处理的次序能是动态的。在不同的处理阶段，语法分析程序在永久存储中校验，以判断在那时是否有任何元数据类别需要执行。此实施例通过从永久存储中删除类别的实体很容地阻止类别。

图4是示出用于处理一命令的输入参数的过程的逻辑流程图400。在这里，已经开发了cmdlet，且元数据已使用图3中示出的Request Object插入到cmdlet的源文件。cmdlet已被编译并已被登录。在登录期间，类名（即cmdlet名）已写入登录存储器。过程400在块401开始，在那里基于反射的外壳已接收了表示cmdlet的输入（如键入）。基于反射的外壳通过从登录处查找该输入并将键入的输入与一个登录的cmdlet相关联而将输入识别成一个cmdlet。处理进行到块402。

在块402，识别与已识别的cmdlet相关的类。此类也能通过登录处识别。处理在块404处理继续。

在块404，读出与该类相关的元数据。元数据包括任何与cmdlet相关的控制指令。控制指令能应用到cmdlet本身或在Request Object中规定的一个或多个参数。在cmdlet登录期间，登录程序将元数据登录到永久存储器中。元数据能以

串行化格式存入XML文件或外部数据库等。如在上面表中所示，在元数据中规定的控制指令，每个与一类别相关，如Applicability Directives, Parsing Guideline Directives等。控制指令的每个类别在基于反射的外壳中的不同阶段处理。每个元数据控制指令处理它自己的错误管理。过程在块406处继续。

在块406，根据识别的类例示Request Object。过程在块408处继续。

在块408处，在Request Object上完成反射以获得有关输入参数的信息。基于反射的外壳提供一常用的界面，用于返回反射数据（按需要）给调用者。在上述实施例中反射使用.NET Reflection。过程在块410处继续。

在块410处，应用可应用性(applicability)控制指令（表1）。可应用性控制指令保证在某些机器角色和/或用户角色中使用该类。例如，某些cmdlet只能由Domain Administrators使用。若在一个可应用性控制指令中规定的约束不满足，发生一错误。过程在块412处继续。

在块412处，使用元数据提供智能感觉(intellisense)。在过程的这一点，当未输入整个命令行。然而，基于反射的外壳知道，通过在与该cmdlet相关的Request Object上的反射所允许的输入参数。因此，一旦输入参数的无二意性的部分在命令行上键入，基于反射的外壳通过引擎自动完成输入参数。只要输入参数的部分能无二意性地识别一个输入参数，能发生自动完成。过程在块414处继续。

在块414处，过程一直等待到对cmdlet的输入参数被输入。通常，这是在用户如用按回车键表示命令行结束时发生。过程在块416处继续。

在块416处，应用语法分析导引控制指令，并用输入参数填入Request Object范例。语法分析程序有一组在语法分析时使用的规则，这组规则规定了在Request Object数据结构中规定的文法被转换成对在命令行上的输入参数的语法的方式。例如，给出下面对命令Foo的Request Object说明：

```
class Foo: Request Object
{
string Name;
Bool Rescure;
}
```

命令行语法能是下列中任一个：

```
$Foo-Name:(string)-Rescure:True
```

$Foo-Name:<string>-Rescure True

$Foo/Name(string)

为了产生希望的语法，该组规则能由系统管理员修改。此外，语法分析程序能支持多组规则，所以用户能使用多于一个的语法。因而，在Request Object结构中规定的文法（如string Name和Bool Recurse)驱动了语法分析器。

通常，语法分析控制指令描述，在命令行中输入的参数如何映射到在Request Object中识别的期望的参数。下面例子说明规定位置信息的语法分析控制指令。

class foo: Request Object

{

[Parsing Parameter Position Attribute(0)]

string Host Name

[Parsing Parameter Position Attribute(1)]

string Alias Name

}。

校验输入参数类型以判断是否正确。若输入参数类型不正确，输入参数能强制变成正确。若输入参数类型不正确且不能强制成正确，打印出用法错误。用法错误使用户能觉察期望的正确语法。用法错误能从Documentation Directive 418获取描述语法的信息。一旦输入参数的类型被映射或被验证。，对应成员被填入在Request Object范例中。过程在决策块420处继续。

在决策块420作出判断，任何输入参数是否需要与用户交互。若任何参数需要与用户交互，过程进到块422。否则过程在块424处继续。

在块422，基于反射的外壳与用户交互以获取输入参数。开发者能规定，能借助对Request Object中的参数规定CmdPg Require Interaction控制指令，通过用户的交互获得输入参数。此外，基于反射的外壳能确定，若不是命令行的所有输入参数被输入，则需要用户的交互。只要输入参数，cmdlet本身，和其他设置不拒绝交互，基于反射的外壳将与用户交互以获得必要的输入参数。能使用标志来规定，在用户层，组的层，及企业层上是否允许用户交互。若一个层不允许用户交互，发生错误消息。一旦完成用户交互，过程在块424处继续。

在块424处，引擎在Request Object范例上完成另一次通过，并应用任何余下的控制指令到输入参数。余下的控制指令包括数据生成控制指令，数据验证

控制指令，对象处理控制指令，和对象编码控制指令。一旦语法分析程序结束分析输入参数，在引擎中处理这些控制指令。现将描述来自在上面表3-5中示出的每个类别的代表的控制指令。

第一个代表的控制指令来自数据生成控制指令。Request Object能包含下列语句：

[Parsing Default Answer Scr: pt Att$_r$: bute(F: lename,F)]

String Name;

Private Arraylist F;

由于Arraylist F是专用的，语法分析程序不将此说明处理成到该cmdlet的输入参数。替代地，Arraylist F是对由在Data Generation控制指令中规定的服务产生的数据的临时存储。在上面例子中，服务是“Filename”。Filename能是由基于反射的外壳提供的实用程序，或能是第三方函数或实用程序。当引擎遇到上述控制指令，该引擎在登录处识别名为“Filename”的服务。服务的登录在服务的安装期间发生。数据生成（Data Generation）控制指令允许在输入参数上发生单独的处理。例如，服务能在命令行上输入成“A*”的文件名上完成通配符扩展。在第二次通过之前，名字成是象它在命令行中输入的那样包含“A*”。在第二次通过期间文件名服务能定位一组以A开头的文件，并将它们存入Arraylist F。本专业行家理解，Data Generation控制指令不仅提供文件名的通配符扩展，也能能完成用户名，过程等的通配符扩展。此外，Data Greneration控制指令能完成在输入参数上的其他处理。

示例性数据验证控制指令能在Request Object中包括下列语句：

[Validation Set Attribute(“Debug,” “Production”, “Test”)]

[Parsing Parameter Mandatory Attribute]

String Name;

给出上面数据验证控制指令，语法分析程序认识到，名字是强制性输入参数，而字符串必须是Debug，Production或Test。否则，数据验证控制指令将发生错误。错误能使用文档控制指令来提供错误消息。

示例的对象处理控制指令能在Request Object中包括下列语句：

[tolower]

String Host Name;

给出上面对象处理控制指令，对Host Name规定的变量在将Request Object

处理成可执行的cmd let之前转换成小写字符串。

示例性对象编码控制指令能在Request Object中包括下列语句：

[TOIP（Host IP）]

String Host Name;

Private Ipaddr HostIp;

给出上面对象编码控制指令，作为Host Name输入的字符串转换成IP地址。语法分析程序不将HostIP参数处理成输入参数，因为它被说明成Private（专用的）。然而可执行的cmdlet能参考Host IP成员。虽然上面的对象编码控制指令看来不要保存许多行代码，实际上，该控制指令大大地减少了由第三方开发者写的代码的量。例如在以前环境中，第三方开发者处理错误过程。此外，若错误消息是字符串，该字符串需要转换成多种语言。然而本发明提供一致的错误处理。而且，本发明提供一机制，用于一致地转换错误消息成多种语言。过程在块426处继续。

在块426，Request Object范例被传送到可执行的cmdlet。然后可执行cmdlet在它的处理中使用这些输入参数。因此，与以前的命令实现相反，本发明的cmdlet不必写任何单独的代码来语法分析或验证输入参数。此外，本发明提供更加丰富的环境，其中软件开发者能通过控制指令规定输入参数的另外处理。

每个控制指令处理它自己的错误消息。若需要显示错误消息，该控制指令从文档存储得到本地化的字符串，并调用显示界面来显示错误消息。有关文档控制指令的数据能存入cmdlet，XML存储或外部数据库，或源文件。因此，本发明能对所有cmdlet提供一致性为错误消息，并减少第三方的本地化工作。

此外，基于反射的外壳提供常见的界面，它返回由以对象的形式的一个控制指令产生的数据到调用的控制指令。调用的控制指令负责将数据塑造成需要的数据类型，虽然表1-5示出若干类别的代表控制指令，能加入其他控制指令，且能加入另外的类别而不偏离本发明的范围。

因此如上所述，本发明提供了为对cmdlet的输入参数定义方法的机制。该机制使开发者能开发，测试和支持cmdlet。该cmdlet具有较少的代码行，较快实现，具有较少缺陷，并容易寻找缺陷。在语法，语义，错误处理，资源管理，安全性等方面该机制提供更多的一致性。上述讨论在.NET架构内描述了本发明。然而本专业的行家认识到，本发明能在任何提供反射能力的操作系统中实现。此外，本发明在命令行的实施例中描述。然而，所分析的可语法分析的数

据能通过语音，图象用户界面，脚本等获得而不偏离本发明的范围。而且本发明能用于通过允许一个Request Object类说明另外的Request Object类来描述丰富的方法组。因此，本发明对系统管理员提供了很大的通用性。

上述说明，例子和数据提供本发明的成分的制作及使用的完全描述。因为能作出本发明的许多实施例而不偏离本发明的精神和范围，本发明归属于附后的权利要求中。

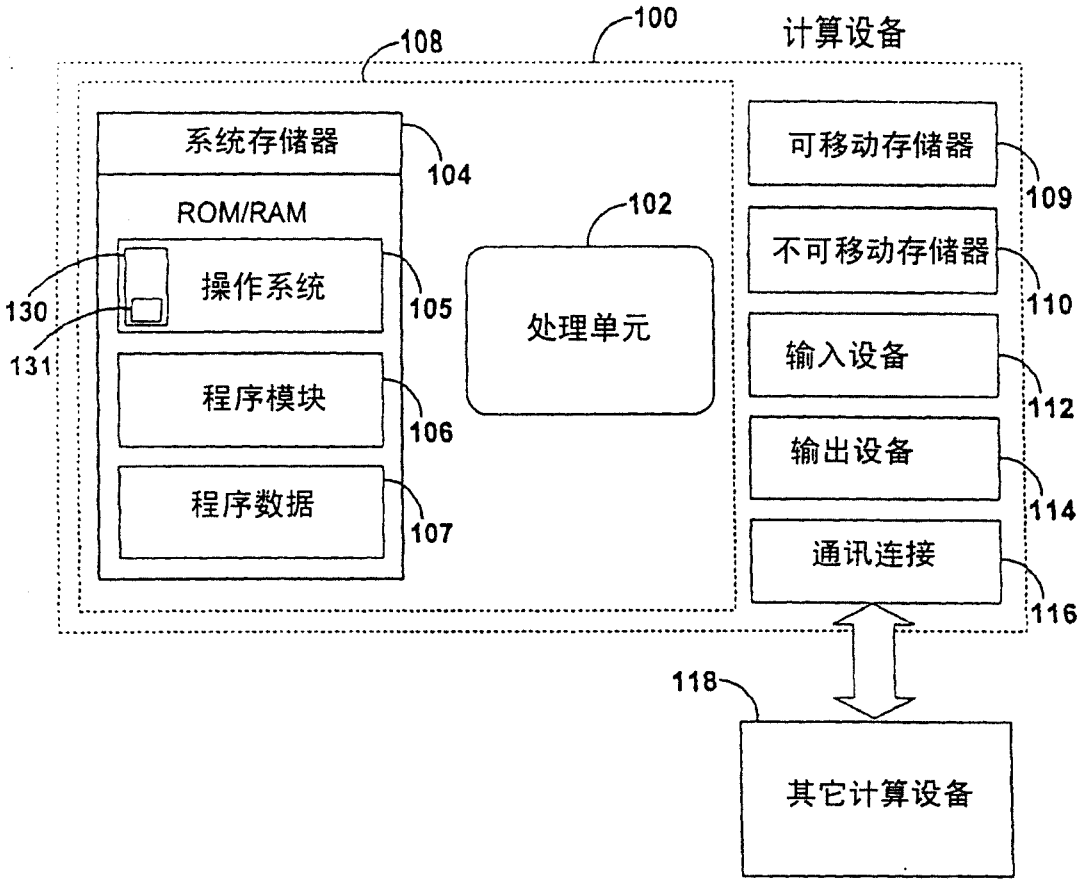

100
计算设备
108
系统存储器
104
ROM/RAM
操作系统
105
130
131
程序模块
106
程序数据
107
102
处理单元
可移动存储器
109
不可移动存储器
110
输入设备
112
输出设备
114
通讯连接
116
118
其它计算设备

图 1

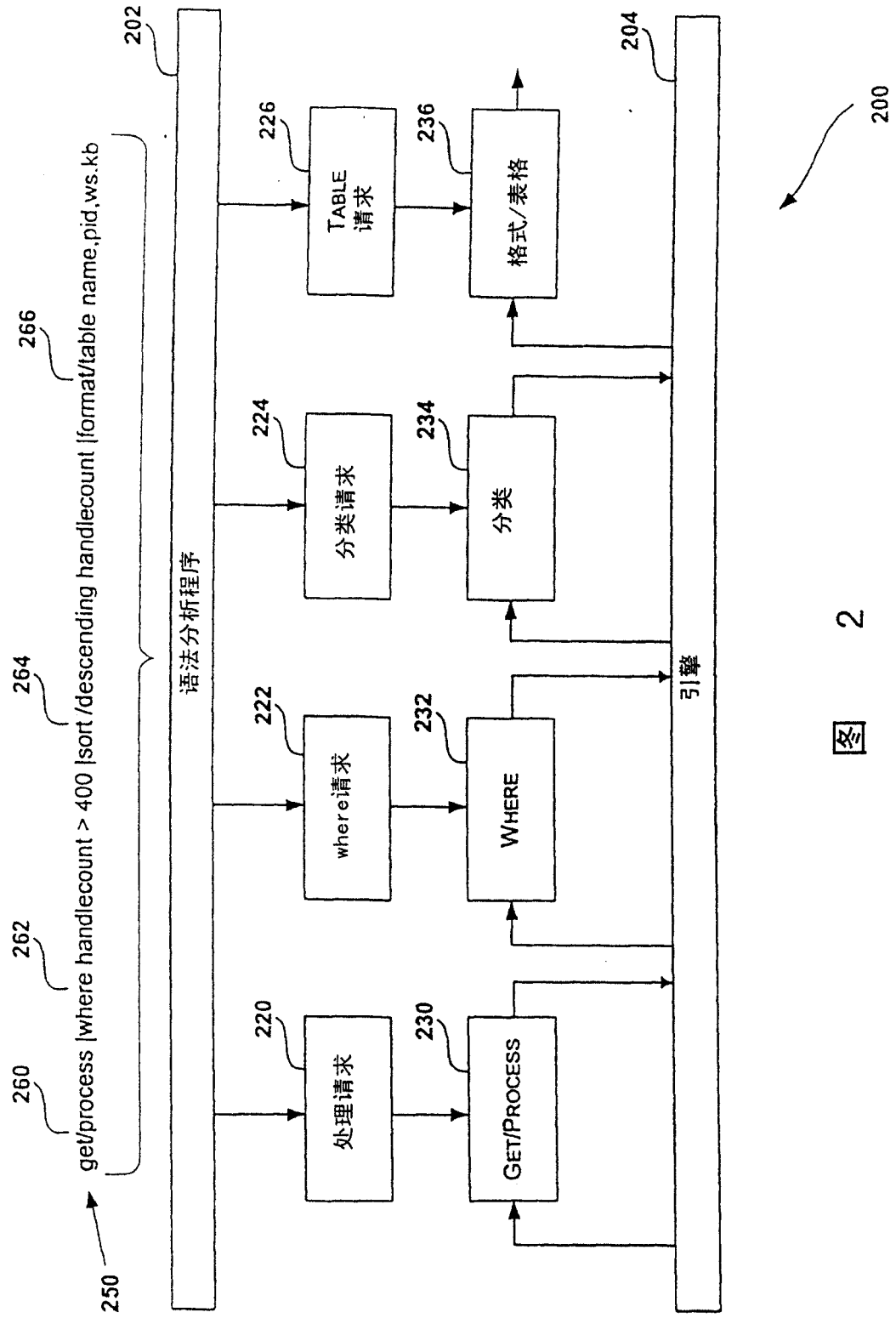

250
260
get/process |where handlecount > 400 |sort /descending handlecount |format/table name,pid,ws.kb
262
264
266
202
语法分析程序
220
处理请求
230
GET/PROCESS
222
where请求
232
WHERE
224
分类请求
234
分类
226
TABLE
请求
236
格式/表格
204
引擎
200

图 2

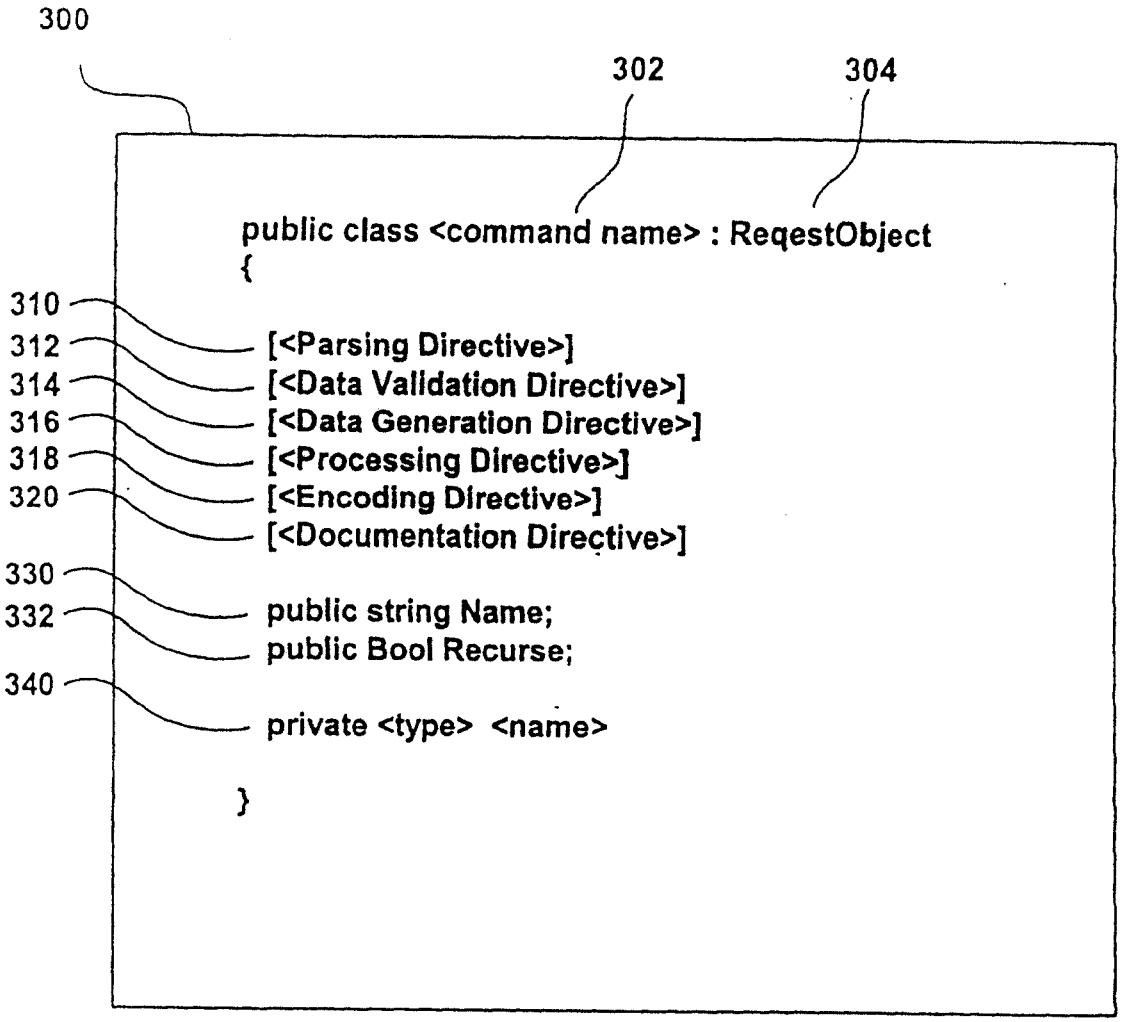

300
302
304
public class <command name> : ReqestObject
{
310
[<Parsing Directive>]
312
[<Data Validation Directive>]
314
[<Data Generation Directive>]
316
[<Processing Directive>]
318
[<Encoding Directive>]
320
[<Documentation Directive>]
330
public string Name;
332
public Bool Recurse;
340
private <type> <name>
}

图 3

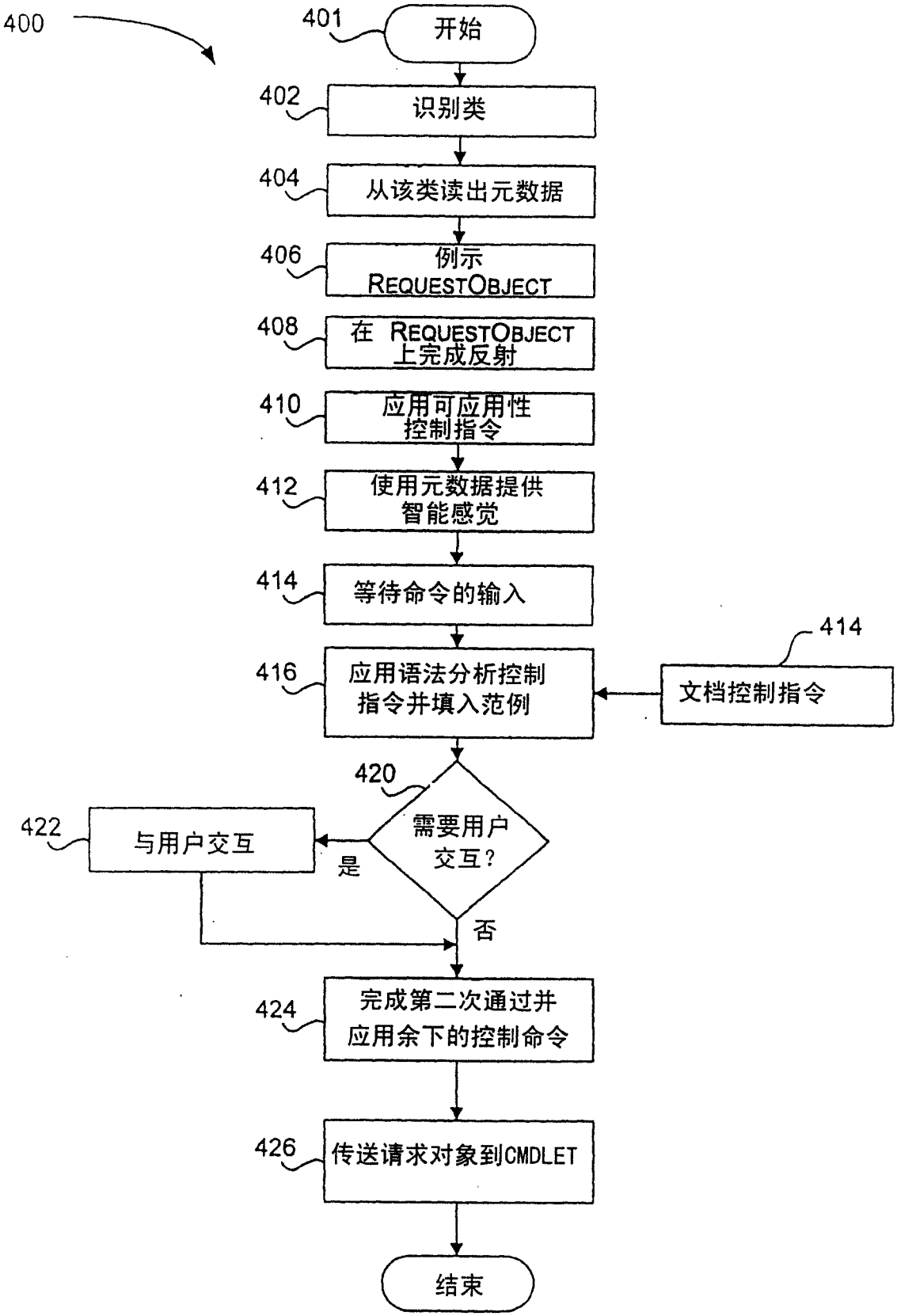

400
401
开始
402
识别类
404
从该类读出元数据
406
例示
REQUESTOBJECT
408
在 REQUESTOBJECT
上完成反射
410
应用可应用性
控制指令
412
使用元数据提供
智能感觉
414
等待命令的输入
414
416
应用语法分析控制
指令并填入范例
文档控制指令
420
422
与用户交互
需要用户
交互?
是
否
424
完成第二次通过并
应用余下的控制命令
426
传送请求对象到CMDLET
结束

图 4