〔19〕中华人民共和国专利局

[11]公开号 CN 1039912A

〔12〕发明专利申请公开说明书

[21] 申请号 89102124.8

[51] Int.Cl[5]

G06F 9/44

〔43〕公开日 1990年2月21日

[22]申请日 89.4.10

[30]优先权

[32]88.7.27 [33]US [31]225,115

[71]申请人 惠普公司

地址 美国加利福尼亚州

[72]发明人 格伦·斯特恩斯 巴巴拉·B·帕卡德 罗尔夫·托马斯·沃森

[74]专利代理机构 中国专利代理有限公司

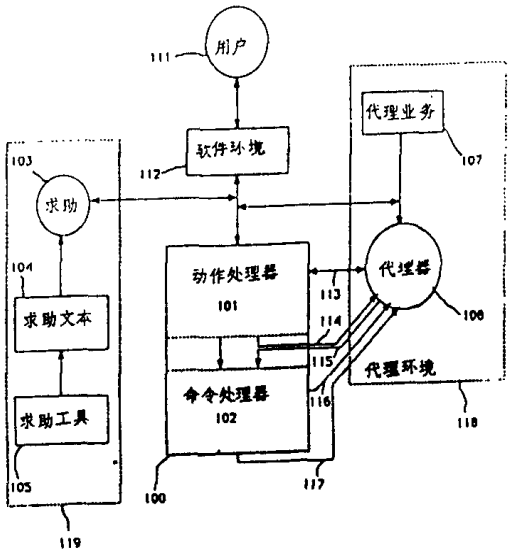
代理人 马铁良 程天正

说明书页数: 25 附图页数: 17

[54]发明名称 一种向用户提供培训的用于多种计算机应用程序的软件代理

[57]摘要

本发明提出了一种计算机系统，它包括一个应用目标程序，一个以计算机为基础的培训程序（"INSTRUCTION 目标程序"）和一个代理器。INSTRUCTION 目标程序与应用目标程序同时运行。应用目标程序包括一个第一动作处理器和一个第一命令处理器。第一动作处理器接收说明由用户采取的语法动作的信息，并根据语法动作产生语义命令。第一命令处理器从第一动作处理器接收语义命令并执行语义命令。



权　利　要　求　书

1、一种计算系统，它包括：

一个监视器，

一个在监视器上显示信息的第一应用目标程序，它对用户采取的语法动作作出响应，第一应用目标程序包括：

第一动作处理器，用于接收表示用户采取的语法动作的信息和按语法动作产生语义命令；

第一命令处理器，用于从第一动作处理器接收语义命令并执行该语义命令；

一个对用户采取的语法动作作出响应的 INSTRUCTION 目标程序，它在监视器上显示有关第一应用程序运行的信息；

一个代理器，它与第一应用目标程序和 INSTRUCTION 目标程序相连接，用于向 INSTRUCTION 目标程序发送语义命令以便向该 INSTRUCTION 目标程序说明在监视器上显示那个信息，以及截听从第一动作处理器向第一命令处理器发送的语义命令。

2、一种按照权利要求1的计算系统，其中 INSTRUCTION 目标程序包括：

INSTRUCTION 动作处理器，用于接收表明由用户采取的语法命令的信息并按语法动作产生语义命令；

INSTRUCTION 命令处理器，用于从 INSTRUCTION 动作处理器和代理器接收语义命令并该执行语义命令。

3、在一种具有一个监视器、一个应用目标程序、一个 INSTRUCTION 目标程序和一个代理器的计算系统中，一种向用户提供以计算机为基础的培训方法，该方法由以下步骤组成：

（a）运行应用目标程序；

（b）把语义命令自代理器发送至INSTRUCTION目标程序，此命令控制INSTRUCTION目标程序以显示有关该应用程序的信息；

（c）利用INSTRUCTION目标程序，显示有关此应用目标程序的信息；以及，

（d）在应用目标程序执行语义命令之前，利用代理器截听由应用目标程序产生的语义命令。

4、按照权利要求3的方法还包括下述步骤：

（e）将由应用目标执行的语义命令从代理器发送至应用目标程序。

5、按照权利要求3的步骤还包括下述步骤：

（f）在INSTRUCTION目标程序执行语义命令之前，用代理器截听由INSTRUCTION目标程序产生的语义命令。

6、在一种有一个监视器和一个代理器的计算机系统中，一种提供演示的方法，它包括操纵在一个第一目标程序的窗口内所显示的一个个图形接口单元，这种方法包括下述步骤：

（a）从代理器发送一个询问信息至第一目标程序，要求第一目标程序向代理器发送有关图形接口单元的信息；

（b）从第一目标程序向代理器发送此信息；以及，

（c）进行演示。

7、按照权利要求6的一种方法，其中，询问信息包括图形接口单元的同一性，以及此信息包括图形接口单元在窗口的位置。

8、按照权利要求6的一种方法，其中，询问信息包括图形接口单元的同一性，以及此信息包括由图形接口单元所表示的第二目标程序的状态信息。

9、按照权利要求6的一种方法，其中，询问信息包括图形接口单元的位置，以及此信息包括图形接口单元的同一性。

10、按照权利要求6的一种方法，其中，步骤（C）包括如下子步骤：

（C1）从代理器向INSTRUCTION 目标程序发送信息，命令INSTRUCTION 目标程序显示对话数据；以及，

（C2）由INSTRUCTION 目标程序在监视器上显示对话数据。

说　　明　　书

一种向用户提供培训的用于多种计算机应用程序的软件代理

本发明属于以计算机为基础的培训（ＣＢＴ），更具体地说，属于应用一个软件代理向用户提供培训。

目前在市场上已有许多ＣＢＴ应用程序，它们补充或代替手册、书面辅导材料和其它普通的说明材料。ＣＢＴ可以是交互式的，且常以一个辅导教师为模型，提供特定的反馈以对培训课期间用户的表现作出响应。

一般来说，ＣＢＴ有两种实施方案。在“模拟的ＣＢＴ”中，应用程序是由一个ＣＢＴ程序所模拟；在“同时的ＣＢＴ”中，应用程序则与一个ＣＢＴ程序同时运行。

随着应用程序变得越来越复杂和更多地运用图形，使用模拟的ＣＢＴ变得更为困难，这是因为一个应用程序的复杂性一般要求模拟此应用程序的程序的复杂性。

同时的ＣＢＴ常比模拟的ＣＢＴ简单，这是因为在同时的ＣＢＴ中，应用程序本身在培训作业期间提供其自己的接口和功能。在一次同时的ＣＢＴ作业期间，ＣＢＴ程序一般将启动应用程序，围绕着应用程序起着一种“外壳”的作用。

在同时的ＣＢＴ作业期间，ＣＢＴ将打开应用程序，并控制它使此应用程序达到一种已知的所需状态。ＣＢＴ应用其自己的子程序，通过在应用程序顶部画出的“窗口”向用户提供培训用正文和图形。正文和图形解释此应用程序的概念并提示用户作出响应。ＣＢＴ监视

用户的输入，以确定用户是否正确地作出响应，还监视显示器屏幕以确定何时应用程序已结束对输入的处理。然后，根据用户作出的响应，CBT再使培训进展下去。

通常，CBT是在语法级上控制和监视应用程序的活动。这里所谓的“语法级”指的是用户为了与一个应用程序对话所进行的动作，如敲键或鼠标的动作。例如，在一个工作于语法级上的CBT中，其中一个应用程序受一个键盘和鼠标所控制，并且输出至一个阴极射线管监视器，CBT将检测出键和鼠标输入以及阴极射线管上象素的状态。这种程度的交互作用称为“语法的”，因为在此级别上，计算机并不从语义上解释与此动作有关的意图。

按照本发明的优选实施例，提出了一种计算机系统，它包括一个应用目标程序，一个以计算机为基础的培训指令目标程序（“INSTRUCTION 目标”）和一个代理器。INSTRUCTION 目标程序与应用目标程序同时运行。应用目标程序包括一个第一动作处理器和一个第一命令处理器。第一动作处理器接收表示用户所采取的语法动作的信息，并根据此语法动作产生出语义命令。第一命令处理器从第一动作处理器接收语义命令并执行语义命令。

INSTRUCTION 目标程序通过语法动作接收用户的输入，并在监视器上显示信息。此信息告诉用户关于第一应用程序的工作。在该优选实施例中，INSTRUCTION 目标程序包括一个INSTRUCTION 动作处理器和一个INSTRUCTION 命令处理器。INSTRUCTION 动作处理器接收表明由用户所采取的语法动作，并根据此语法动作产生语义命令。INSTRUCTION 命令处理器从INSTRUCTION 动作处理器接收语义命令并执行语义命令。

操作一任务语言程序的软件代理，向INSTRUCTION目标程序发出语义命令，指示INSTRUCTION目标程序显示哪些信息。此软件代理还监视应用目标程序和INSTRUCTION目标程序，在语义命令被执行之前截听这些命令。

图1为一个方块图，说明在一个应用程序、一个软件代理环境和一个求助环境之间的相互关系。

图2为一个方块图，说明按照本发明的优选实施例，一个任务语言文件是如何产生和执行的。

图3为一个方块图，表示按照本发明的优选实施例在图1所示的应用程序。

图4为一个方块图，表示按照本发明的优选实施例，通过图1所示的应用程序的数据流。

图5为一个方块图，表示按照本发明的优选实施例的编译程序。

图6表示按照本发明的优选实施例的一个计算机、监视器、键盘和鼠标。

图7为图6所示的鼠标的顶视图。

图8为图5所示的编译程序中的数据流。

图9为一个方块图，它表示把一个执行以计算机为基础的培训的应用程序加至图1所示的应用程序和软件代理环境。

图10～17表示，由于图9所示的系统执行一项任务的结果，在图6所示的监视器上显示的内容。

图1所示的为根据本发明的优选实施例的一种计算系统的方块图。用户111通过一个软件环境112与计算机系统通信。例如，软件环境112可以是Microsoft Windows（微软件窗口），即由

Microsoft 公司所出售的一种程序，该公司的营业地址为 16011 NE 36th Way , Redmond , WA 98073-9717 。软件环境１１２与一个应用程序１００相互作用。含有描述用户动作的信息被软件环境１１２送至应用程序１００。在该优选实施例中，含有用户动作的信息为由 Microsoft Windows 发送的标准信息。应用程序１００含有一个动作处理器１０１，它将用户的语法动作转换成单个的语义命令。例如动作处理器１０１观察和收集用户的动作，如击键和由用户使用的鼠标的运动。只要用户的动作符合一个命令的语法，则产生一个语义命令。有多种可为用户动作使用以产生单个的语义命令的方式。用户产生语义命令的方式可以不同，但语义命令的执行总是相同的。动作处理器１０１能从语法上说明用户用于构成一个特定的语义命令的许多方式。除用户的语法动作外，动作处理器１０１还处理输至应用程序１００的其它信息。其中有些信息将会导致及产生一个语义命令；另一些则将完全由动作处理器１０１进行处理。

应用程序１００还包括一个命令处理器１０２，它执行语义命令。命令处理器１０２接收内部形式的语义命令（内部形式在下文中还将详细介绍），如果一个命令不能被执行，则回送一个出错。

应用程序１００和软件环境１１２在它们之间的接口层与求助环境１１９相互作用。求助环境１１９含有一个求助应用程序１０３，它使用一个求助文本１０４。求助环境１１９还包括一个求助工具１０５，它用于产生求助文本１０４。

软件环境１１２还与一个代理环境１１８相互作用。代理环境１１８包括一个代理任务１０７和一个代理器１０８。

代理器１０８与应用程序１００在五种不同的概念类别上相互作

用以执行五种功能。代理器１０８与动作处理程序１０１通过一个数据通道１１３相互作用进行询问。代理器１０８通过一个数据通道１１４在动作处理器１０１和命令处理器１０２之间相互作用，以监视应用程序１００的活动。代理器１０８通过一个数据通道１１５与命令处理器１０２相互作用，以使命令由应用程序１００执行。代理器１０８通过一个数据通道１１６与命令处理器１０２相互作用，以处理在应用程序１００中处理命令过程中的出错。代理器１０８通过一个数据通道１１７与命令处理器１０２相互作用，以记录应用程序１００的执行和接收一个命令结束的通知。

在本发明的优选实施例中，命令可以用四种方式表达：（１）任务语言的形式，以关键字和参数贮存；（２）p码形式，它是由代理器１０８翻译的外部形式的二进制码，并带有附加的标题；（３）外部形式，它是为应用程序所理解的二进制数据，并在代理器１０８和应用程序１００之间传送；（４）内部形式，为二进制命令，这些命令在应用程序１００中执行。这四种表示命令的方式在本文的附录A中将有进一步的介绍。

图２为说明整个代理系统工作的方块图。任务语言文件１３１是一个含有任务语言的文件。任务语言是描述一个应用程序功能的命令的文本形式。任务语言由类别相关的命令和类别无关的命令所组成。类别相关的命令为由一个应用程序执行的命令。在图２中仅示出一个应用程序１００。但是，代理器１０８可与许多应用程序相互作用。

在本发明的优选实施例中，由应用程序操作的数据文件通过使用目标程序来进行访问。每个目标程序包括访问一个数据文件和访问一个应用程序。那些引用相同的应用程序的目标程序被称作是同一类别

的成员。每个应用程序执行一组不同的命令。因此，类别相关的命令对不同的应用程序是不一样的。

代理器108执行类别无关的命令，这些命令为代理器108所能理解。类别无关的命令由代理器108执行，而不是由应用程序所执行。

任务语言文件131由一个类别无关的分析程序122用于准备一个p码文件121。在准备p码文件121过程中，类别无关分析程序122调用类别相关分析程序123、124等。如下文还将进一步介绍的，一个类别相关的分析程序是这样的一个分析程序，它产生类别相关的命令，这些命令以p码形式封装。代理器108从p码形式提取其外部形式的命令，将这些命令送往适当的应用程序。p码内的一个类别域表明哪一个应用程序应接收一个特定的类别相关的命令。类别无关分析程序122是这样一个分析程序，它产生由代理器执行的p码。

任务语言文件131可由用户111用一个代理任务编辑程序132来准备。或者，任务语言文件也可用一个类别无关的记录装置125来准备，该记录装置则利用类别相关的记录装置126、127等。总的说来，记录装置记录应用程序的命令供以后读出之用。当计算系统处在记录工作方式时，代理任务编辑程序132从应用程序（如图示的应用程序100）接收输入，它详细说明代理器108和应用程序将采用什么动作。应用程序通过一个应用程序接口（API）130与代理任务编辑程序132连接。当计算系统处于记录工作方式时代理任务编辑程序132把数据输送到类别无关的记录装置125，当代理任务编辑程序被用户111所用时，则输送到

任务语言文件１３１。

类别无关的记录装置１２５接收此信息并构成任务语言文件１３１。当类别无关的记录装置１２５发现代理任务编辑程序１３２正在输送有关被一个应用程序所采取的动作的信息时，类别无关的记录程序就调用该应用程序的类别相关的记录程序，然后此记录程序产生用于该动作的任务语言形式。类别无关的记录程序１２５产生由代理器１０８采取的行动的任务语言形式。

在执行ｐ码文件１２１时，代理器１０８读取每个ｐ码命令，并确定此ｐ码命令是否会有一个将由代理器１０８执行的类别无关的命令，或一个由一个应用程序执行的类别相关的命令。如果ｐ码命令含有一个类别无关的命令，则代理器１０８执行此命令。如果ｐ码命令含有一个类别相关的命令，代理器由ｐ码命令确定将接收此命令的应用程序。然后代理器１０８提取一个插入在ｐ码中的外部形式表示的类别相关的命令。此类别相关的命令再被送至该应用程序。例如，如果类别相关的命令是用于应用程序１００，则此类别相关的命令被送至应用程序１００。在应用程序１００内，一个译成内部形式的装置１２８被用来将类别相关的命令（以外部形式传送的）翻译至命令的内部形式。

ＡｐＩ１３０用于在代理器１０８和应用程序１００之间作为接口。ＡＰＩ１３０为一组函数和信息，用于访问代理器１０８和其它部份。

当系统处于记录工作方式时，译成内部形式的装置128对来自代理器１０８的命令进行翻译，然后通过图3中的一个命令接口部件１４６将它们输送到命令处理器１０２。一个译成外部形式

的装置１２９接收已被命令处理器１０２执行的内部形式的命令。该命令通过回送接口部件１４７所接收，如图３所示。译成外部形式的装置１２９将内部形式的命令译成外部形式的命令。然后外部形式的命令再通过ＡＰＩ１３０传送至任务编辑程序１３２。

图３示出的是本发明的优选实施例中应用程序１００的详细组成。应用程序１００包括一个用户动作接口单元１４５，它与软件环境１１２和命令接口单元１４６相互配合，命令接口单元１４６则与动作处理器１０１和命令处理器１０２连接。如图所示，动作处理程序１０１和命令处理程序１０２都访问应用程序数据１４４。一个回送接口单元１４７对命令处理器１０２作出响应，并把控制回送至软件环境１１２。图示的译成外部形式的装置１２９与回送接口单元１４７相互作用。只有在应用程序１００处于读出或记录工作方式时，回送接口单元１４７才被调用。这些工作方式在下文中还将充分阐述。回送接口单元１４７向代理器１０８指明一个命令已被应用程序１００所执行，应用程序１００已准备好执行下一个命令。

在应用程序１００内还有一个模态对话区处理器１４８和一个出错对话区单元１４９。这两部份都与软件环境１１２相互作用以控制对话区的显示，对话区则与用户１１１通信。

一些应用程序能同时在一个以上的窗口工作。当这样工作时，对应用程序在其中工作的多个窗口的每一个加有一个无模式用户动作接口单元、一个无模式动作处理器和一个无模式命令接口单元。例如，在应用程序１００中示出有一个无模式用户动作接口单元１４１、一个无模式动作处理器１４２和一个无模式命令接口单元１４３。

图４所示的是应用程序１００内的数据流。送给应用程序１００

的信息由一个用户动作接口单元１４５所接收。对于某些类型的信息，如来自求助应用程序１０３的信息，用户动作接口单元１４５使应用程序１００立即返回。否则，此信息将送至读出信息测试单元１５０。

如果此信息是用于读出已由记录或分析而产生的命令，则此信息将送至译成内部形式的装置１２８，１２８再将此信息中的命令由外部形式转换至内部形式。然后此命令再输至命令接口单元１４６。

如果此信息不是一个读出信息，则此信息被送至动作处理器１０１，以便（例如）从语法上解释引起产生此信息的一个用户动作。如果没有由动作处理器１０１或由内部处理程序１０８产生的语义命令，则读出信息测试单元１５０使得应用程序１００返回。如果有一个所产生的语义命令，则此命令被送至命令接口单元１４６。

如果代理器１０８正在监视应用程序１００执行命令，则命令接口单元１４６将所接收到的任何数据送至译成内部形式的装置１２９，该装置１２９把命令译成外部形式，再将命令传输至代理器１０８。命令接口单元１４６还将数据送至一个模态对话区测试单元１５２。

如果被传送的数据含有一个用于对话区的请求，则模态对话区测试单元１５２把数据送至模态对话区处理器１４８进行处理。否则模态对话区测试单元１５２将此数据送到命令测试单元１５１。

如果数据含有一个命令，则命令测试单元１５１将该命令送至命令处理器１０２执行。命令测试单元１５１把数据送往回送接口单元１４７。

如果代理器１０８正在记录命令，回送接口单元１４７把数据送

至译成外部形式的装置１２９以便转换为外部形式，并通过回送接口单元１４７传送至代理器１０８。回送接口单元在下一个信息收到时返回。

图５示出的是在任务语言编译程序１２０中的数据流。任务语言编译程序１２０用来产生ｐ码文件１２１，它由用户用任务语言编写。任务语言文件１３１包含有用户所写的命令。在本发明的优选实施例中，任务语言是按照本文的附录Ｂ中的代理任务语言准则来编写的。

任务语言编译程序１２０是一个两遍编译程序。在第一遍时，所用的程序包括有一个输入流处理器１６４、一个表达式分析程序１６６、一个类别相关的分析程序１２２、一个保存文件缓存器１７１、第二遍程序１７４和类别相关的语法分析程序（所示出的有类别相关语法分析程序１２３、一个现行的类别相关的语法分析程序１６７和一个类别相关的语法分析程序１６８）。由于第一遍扫描的结果，建立起一个暂时文件１７６。

类别无关的语法分析程序１２２分析类别无关的任务语言命令。在系统内运行的每个应用程序还有它执行的特殊命令。因此，对每个应用程序都开发了一个单独的类别相关的语法分析程序。此语法分析程序能分析它为之开发的应用程序将执行的命令。随着应用程序加入系统或从系统删去时，类别相关的分析程序可加入到任务语言编辑程序１２０或从其中删去。

此外，图上还示出有一个ＣＢＴ语法分析程序１２５。当执行ＣＢＴ时，ＣＢＴ语法分析程序被用来分析由代理器１０８运行而产生的码。

当编译开始时，类别无关的语法分析程序１２２从输入流处理器

１６４要求一个标记。输入流处理器１６４对任务语言文件１３１进行扫描并产生标记。然后，类别无关的语法分析程序１２２完成下列几件事之一。类别无关的语法分析程序１２２可能产生送至保存文件缓存器１７１的ｐ码。如果类别无关的语法分析程序１２２期待的下一个标记是一个表达式，则类别无关的语法分析程序１２２将调用一个程序Makeexpression（完成表达式）(　)，它调用表达式语法分析程序１６６。表达式语法分析程序１６６从输入流处理器１６４请求标记直至表达式结束。然后表达式语法分析程序１６６产生ｐ码以送至保存文件缓存器１７１，再保存在暂时文件１７６中。此外，表达式语法分析程序１６６产生一个表达式标记，此记号被回送至输入流处理器１６４。输入流处理程序１６４在类别无关的语法分析程序１２２要求此表达式时，将此表达式送至类别无关的语法分析程序１２２。

由于执行一个Focus 命令的结果，一个特定的类别相关的语法分析程序将具有优先权。因此，在它的语法分析环中，类别无关的扫描程序１２２ａ将调用当时有Focus 命令的应用程序的类别相关的语法分析程序。此类别相关的语法分析程序将从输入流处理程序１６４请求标记，直至它接收到一个类别相关的命令，从而由类别相关的语法分析程序调用的语义程序转换为外部命令形式，或直至类别相关的语法分析程序确定它不能分析所接收到的表达式。如果类别相关的语法分析程序遇到一个表达式，它可通过调用 Makeexpression (　)来调用表达式语法分析程序１６６。如果类别相关的语法分析程序不能分析它所接收到的标记，则类别相关的语法分析程序回送一个出错，类别无关的语法分析程序将试图分析这些标记。

一个Focus OFF 命令将导致使类别无关的语法分析程序122分析所有的命令，而不将它们送往一个类别相关的语法分析程序。当分析一组类别无关的命令时，这样做可避免类别相关的语法分析程序软件不必要的运行，从而节省为编译任务语言所需的计算时间。

CBT编译程序指令使得CBT编译程序的一个标志为“通”或“断”。CBT编译程序标志确定CBT语法分析程序125是否可被调用来分析命令。分析的优先次序将在下文讨论。

命令将先将送至有Focus 命令的任何类别相关的语法分析程序。如果没有具有Focus 命令的类别相关的语法分析程序，或者如果具有Focus 命令的类别相关的语法分析程序不能分析此命令，则只要CBT编译程序的标志为“通”，该命令将送到CBT语法分析程序125进行分析。如果CBT编译程序的标志为“断”，或者CBT语法分析程序125不能分析此命令，则此命令将被类别无关的语法分析程序122所分析。

图6所示为一个INSTRUCTION 目标程序可以运行的计算机18。还出出有一个监视器14，一个鼠标20和一个键盘19。图7所示为带有按钮27、28的鼠标20。

图8为类别无关的语法分析程序122和类别相关的语法分析程序之间的数据流。类别相关的分析程序示出的是语法分析程序123和124。为了集中讨论语法分析程序之间的关系，在图8的说明中，扫描程序122a对表达式语法分析程序166的调用不予考虑。

同样，图上画出CBT语法分析程序125和CBT语法分析程序125的类别相关扫描程序125a。当CBT标志为“通”时，如上所述，分析程序的优先顺序为带有Focus 命令的类别相关的语

法分析程序，然后是CBT语法分析程序，最后为类别无关的语法分析程序122。在以下的讨论中，为便于说明，假定CBT标志为“断”。

当类别无关的语法分析程序122准备好接收一个标记时，它调用一个扫描程序122a。扫描程序122a检查是否有一个Focus命令相对一个应用程序。如果在一个应用程序中没有一个Focus命令扫描程序122a就调用输入流处理程序164，它向扫描程序122a回送一个标记。扫描程序122a再将此标记回送给类别无关的语法分析程序122。

如果在一个应用程序中有一个Focus命令，该应用程序的类别相关的语法分析程序具有优先权并被调用。例如，当有Focus于语法分析程序123的应用程序时，语法分析程序123通过一个类别相关的扫描器123a调用扫描程序122a。扫描程序122a检查它的状态，并确认它正在被一个类别相关的语法分析程序所调用，因此它不递归地调用另一个类别相关的语法分析程序。扫描程序122a调用输入流处理程序164，它向扫描程序122a回送一个标记。扫描程序122a通过类别相关的扫描程序123a向类别相关的语法分析程序123回送此标记。虽然本发明的执行过程中包括有相关的扫描程序123a，在其它的执行过程中也可能没有相关的扫描程序123a，语法分析程序123可直接调用扫描程序122a。

只要相关的语法分析程序123能分析它所接收的标记，相关的语法分析程序123就能继续通过相关的扫描程序123a请求标记。用这些标记，相关的语法分析程序将调用语义程序，后者将产生插入

在p码中的类别相关的外部指令。当相关的语法分析程序１２３不能分析它所接收的一个标记时，相关的语法分析程序将向扫描程序１２２a回送一个出错。扫描程序１２２a然后再调用输入流处理程序１６４并从它接收相关的语法分析程序１２３不能分析的标记。此标记被回送到无关的语法分析程序１２２。无关的语法分析程序122分析此标记，调用语义程序来产生p码供代理器１０８执行。下一次无关的语法分析程序１２２从扫描程序１２２a请求一个标记，扫描程序１２２a将重新调用相关的语法分析程序１２３直至存在一个Focus OFF 命令或在另一个应用程序上有一个Focus 命令。

当Focus 命令是对相关的语法分析程序１２４的应用程序上时，扫描程序１２２a将调用相关的语法分析程序１２４。相关的语法分析程序１２４调用一个相关的扫描程序１２４a并按类似于相关的语法分析程序１２３进行工作。

图5中所示的保存文件缓存器１７１从类别无关的语法分析程序１２２和从表达式分析程序１６６接收p码，也从类别相关的语法分析程序接收插入p码中的外部形式命令。保存文件缓存器１７１将此信息贮存在一个暂时文件１７６中。第二遍程序１７４接收贮存在暂时文件１７６中的这个p码和外部形式命令并完成内务操作，如确定地址等，以便产生p码文件１２１。

在图9中，应用程序１００被包含于（例如）一个目标程序“NewWave Office”中。窗口３００为用户与目标程序“NewWave Office”之间的接口。为了培训用户如何使用目标程序“NewWave Office”，一个INSTRUCTION 应用程序２００与应用程序１００同时运行。INSTRUCTION 应用程序200

被包含在一个INSTRUCTION 目标程序中，INSTRUCTION 应用程序２００与其它应用程序或系统在设计上相似。如图所示，INSTRUCTION 应用程序２００有一个动作处理器２０１和一个命令处理器２０２。

代理器１０８与INSTRUCTION 应用程序２００的相互作用与系统中其它应用程序相同。例如，代理器１０８通过一个数据通道２１３与动作处理器２０１相互作用进行询问。代理器１０８通过一个数据通道２１４在动作处理器２０１和命令处理器２０２之间相互作用，以便监视应用程序２００的活动。代理器１０８通过一个数据通道２１５与命令处理器２０２相互作用以便使命令为INSTRUCTION 应用程序２００所执行。代理器１０８通过一个数据通道２１６与命令处理器２０２相互作用以便处理在INSTRUCTION 应用程序２００中执行一项命令的出错。代理器１０８通过一个数据通道２１７与命令处理器２０２相互作用，以便记录INSTRUCTION 应用程序２００的执行和接收一个命令结束的通知。

INSTRUCTION 应用程序２００在执行应用程序１００的命令时与用户相互作用，例如通过显示对话区（如窗口３０２）的方式进行。INSTRUCTION 应用程序２００还可通过使用其它方法（如话音）来通信。

图１０～１７表示的是一个简明的ＣＢＴ对话过程。在对话期间，用户被告知如何打开一个档案文件“Fred”，在图１０中它用一个图象３０１来表示。在图１０上还示出一个图象３０４，它表示代理器１０８。图象３０９表示一个目标程序“Lesson Task”

（“课程任务”），它包括如下表1所示的任务语言文件的已编译的p码形式。任务语言文件的编译的p码形式被代理器108所运行。如果目标程序“Lesson Task”（“课程任务”）在编译之前已被打开，则可以对任务语言文件的p码形式的源码进行编辑。

一个图象305表示一个被称为“Lesson Instruction"("课程说明”）目标程序，它包括被称为对话数据的数据，含有向用户显示数据的库程序的INSTRUCTION 应用程序200。目标程序“Lesson Instruction"("课程说明”）在被代理器108命令时显示此数据。受鼠标20控制的箭头303在图上指示图象309。

当箭头303处于图象309上时，用户可通过按钮27来选择目标程序“Lesson Task”（“课程任务”）。同时，图象309最亮，如图11所示。图象309的阴影307将跟随着箭头303。当阴影307被置于图象305上、并且按钮27被释放，则图象309将从窗口300消失，如图12所示，而代理器108将开始运行包含在目标程序“Lesson Task”（“课程任务”）中的任务语言程序。下表1中给出的例子是包含在Lesson Task（“课程任务”）中的该任务语言程序的源程序。

表 1

```
1     task
2     cbt on
3     OPEN#       =  1
4     SELECT# =  211
5     focus on office "NewWave Office"
6     select instruction "Lesson Instruction"
7     open
8     focus on instruction "Lesson Instruction"
9     show_window 1
10    on command do process_button
11    button_flag#  = 0
12    set command on
13    while button_flag# = 0
14       wait
```

```
15   endwhile
16   set command off
17   hide_window 1
18   show_window 2
19   on command do process_open
20   open_flag# = 0
21   set command on
22   while open_flag# = 0
23     wait
24   endwhile
25   set command off
26   hide_window 2
27   show_window 3
28   on command do process_button
29   button_flag# = 0
30   set command on
31   while button_flag#  = 0
32     wait
33   endwhile
34   set command off
35   hide_window 3
36   end task
37
38   procedure process_button
39     if sys_cmdclass() = "INSTRUCTION"
40       button_flag#  = 1
41     endif
42     ignore
43   endproc
44
45   procedure process_open
46     if sys_cmdclass() = "INSTRUCTION"
47       do demo
48       open_flag# = 1
49       ignore
50     else
51       cmd#  = sys_command()
52       if cmd# = SELECT#
53         class# = sys_commandparm(1,0)
54         title#  = sys_commandparm(1,len(class#))
55         execute
56       else
57         if cmd# = OPEN# and class# = "FOLDER" and
           title# = "Fred"
58           open_flag# = 1
59           execute
60         else
61           ignore
62         endif
63       endif
64     endif
65   endproc
66
67
```

```
68   procedure demo
69      focus on office "NewWave Office"
70      object_region#  = where_is("FOLDER", "Fred")
71      point to center (object_region#)
72      double_click left button
73      pause 5
74   endproc
```

在表1中的任务语言是通过使用一个类别无关的语法分析程序122、一个用于目标程序“NewWave Office”的类别相关的语法分析程序、一个用于目标程序“Lesson Instruction"("课程说明”)的类别相关的语法分析程序和一个CBT分析程序125，由任务语言编译程序120所编译。例如，第5行中的“Focus”命令由类别无关的语法分析程序122所分析，第6行中的“Select Instruction”(选择指令)命令是由目标程序“NewWave Office”的类别相关的语法分析程序所分析，第9行中的“Show—Wnidow”(显示窗口)命令由目标程序“Lesson Instruction"(“课程说明”)的类别相关的语法分析程序所分析，第71行中的“Point to Center”(指向中心)命令由CBT语法分析程序125所分析。

表1中程序的第1行含有字“task”(任务)，因为每个任务语言程序的第1行都含有指令“task”。同样，表1程序的第36行含有字“end task”(任务结束)，表示这是任务程序中的最后一条指令。

第2行的指令将CBT编译程序标记“通”。第3和4行的指令

设定变量。第5行的指令将Focus 置于目标程序“ NewWave Office”上。第6行和第7行的指令将被代理器108送至目标程序“ NewWave Office ”。当被目标程序“ NewWave Office”执行时，这些指令将使目标程序“ Lesson Instruction"("课程说明”）被选择和打开。当目标程序“ Lesson Instruction"("课程说明”）被打开时，它运行INSTRUCTION 应用程序200。在图13上显示最亮的图象305表明选择了“ Lesson Instruction"（“课程说明”）。

第8行的指令将Focus 命令置于目标程序"Lesson Instruction"（“课程说明”）上。第9行的指令，当由代理器108执行时，将被代理器送往目标程序“ Lesson Instruction"("课程说明”）。当由目标程序“ Lesson Instruction"("课程说明”)执行时，此指令将在窗口300的顶部打开窗口302，如图14所示。窗口302告诉用户关于如何打开一个目标程序。当用户读完窗口302后，他利用鼠标将箭头置于一个按钮308上，再按压按钮27。实质上代理器108等待用户选择按钮308，其中标有“继续”字样。代理器108将使其它每个命令都被忽略。代理器108是通过监视被其它运行的目标程序所执行的应用程序和截听被执行之前的命令实现这一点的。为此所需的程序将在下文中介绍。

当用户读窗口302时，第10～14行中的指令被代理器108所执行。第10行中的指令确定一个监视过程。在第11行中，变量“ Button-flag ”（按钮－标志）为清“零”。第12行的指令“接通命令”接通监视。当监视接通时，在用户执行任何命令时，过程“ Process-button"将被执行。这相应于在一个应用程序中在一

个动作处理器和一个命令处理器之间沿着一条数据通路传送任何命令，例如，沿着应用程序１００的动作处理器１０１和命令处理器１０２之间的数据通道１１４，这称作命令陷阱。因为代理器１０８能监视“NewWave Office”应用程序１００和INSTRUCTION 应用程序２００，故能产生命令陷阱。当代理器产生一个命令陷阱时，从动作处理器１０１送至命令处理器１０２的命令和动作处理器２０１送至命令处理器２０２的命令被代理器１０８所截听，从而使目标程序“NewWave Office”和“课程１”被代理器１０８所监视。从这二个目标程序来的语义命令在执行之前被监听，并形成一个命令陷阱。

过程“Process -button"表示在表1第３８～４３行，函数“Sys-cmdclass( )”（系统命令类别）回送一个从用户接收此命令的目标程序的类别字符串。在第３９行中，目标程序为INSTRUCTION 应用程序２００，也就是说，如果由于用户已把箭头３０３置于按钮３０８上并按压按钮２７使过程“Process button”被调用，则函数“Syo-cmdclass( )”将回送字符串“INSTRUCTION”，并且第４０行中变量“Button-flag#”被置成“１”。在另一方面，如果任何其它的目标程序从用户接收一个命令，例如，如果用户把箭头３０３置于窗口３００并按压按钮２７，则目标程序的类别字符串将被回送且不等于“INSTRUCTION”，并且变量“Button-flag”将不置为“１”。

在第４２行的“Ignore”（忽略）命令表明，不管那个目标程序（即不管是目标程序“NewWave Office”或“课程１”）回

送从用户来的命令，该命令本身被忽略。也就是说，不管命令来自“NewWave Office”应用程序100或INSTRUCTION 应用程序200，该命令不被回送至命令处理器102或命令处理器202作进一步处理。反之，通过把命令设置API－NO－CMD而回送一个NULL（零）命令。

在第12行中接通监视后，代理器执行第13行中的命令，进入一个While （短时）循环（指令13～15），等待用户用箭头303选择按钮308。处于此循环中，当在一个应用程序中产生任何命令时，过程“Process–button"运行。在过程“Process–button"结束时，如果Button–flag# ＝0，代理器108继续处于如第13～15行中的指令所确定的循环中。如果Button–flag＃＝1，代理器108继续执行程序，从第16行的指令开始。

在第16行中，监视被“断”。第17和第18行的指令被代理器108送至目标程序“Lesson Instruction"("课程说明”）以便由INSTRUCTION 应用程序200执行。这二个指令的执行使INSTRUCTION 应用程序200从显示器14上移去窗口302和显示窗口311，如图15所示。第19行的指令被代理器108执行时，重新定义监视器过程为过程“Process–open”。

在第20行，变量“open–flag#”（打开标记）被置为“0”。在第21行，指令“Set Command on”把监视“接通”，使命令在被系统中运行的任何目标程序执行之前，被代理器108所截听。

第22～24行中的指令通知代理器108在继续执行程序之前等待，直至过程“Process–Open”中的变量“open–flag”被

置为“１”。当在一个应用程序中产生一个命令时，由于用户动作的结果，过程“Process—open”被运行。在结束时，如果Open—flag＃＝0，代理器继续处于由第２２～２４行所确定的循环中。如果Open—flag＃＝1，代理器１０８继续执行程序，从第２５行的指令开始。

过程“Process—Open”在第４５～６５行中表示。如果在第４６行中，函数“Sys—cmdclass(　)”回送“INSTRUCTION”，这表示它是曾试图执行一个命令的INSTRUCTION 应用程序２００。这意味着用户已选择按键３１４，请求一个演示。因此在第４７行，过程“demo”（演示）被调用。在第４８行，变量“Open—flag＃”被置“１”。在第４９行，代理器１０８被指示忽略正被监视的命令。

过程“demo”表示在第６８～７４行，它向用户表明如何打开档案文件“Fred”。第６８行的指令把“Focus”置于目标程序“NewWave Office”。询问函数“Where—is”（那里是…）（“档案文件”，“Fred”）询问目标程序“NewWave Office”在它的显示上（即窗口３００）那里是档案文件Fred。当代理器１０８执行此指令时，代理器发送一个API—INTERROGATE—MSG信息给目标程序“NewWave Office”，后者以档案文件Fred在窗口３００的位置座标作出响应。

在第７０行的指令中，变量“Object—neglob”（目标程序范围”被置为由函数“Where—is”(“FOLDER”，“Fred”) 所回送的值。第７１和７２行的指令产生有顺序的用户动作级别的信息，它通过用户动作接口送至目标程序“NewWave、Office”供执行。

如图16所示第71行的指令使箭头303移动以指向档案文件Fred的中心。在图16中，箭头303的运动是由表示在运动途径310的起点和终点上的箭头303所表示。在第72行，一个用户动作（相当于按压鼠标20的按钮27二次）被送至目标程序“NewWave Office”进行处理。在第73行，一个5秒的暂停使用户有机会去对他所看到的作出响应。

总的来说，存在三种类型的询问功能。功能“Where-is( )”具有一个目标程序的类别和标题的标识符作为参数，并要求在监视器14的图象显示器内的长方形区域代表该目标程序。功能“Whats-at( )”（什么在那里( )）有显示器上的一个点作为功能，并要求在该点由一个图象所表示的任何目标程序的同性。功能“Status( )”（状态）具有一个目标程序的标识符作为一个功能，并要求该目标程序的状态，如该目标程序是否有Focns 命令或是否有一个供该目标程序用的打开的窗口。每一个这类功能的指令，当被代理器108执行时，会导致从具有Focus 的目标请求信息的代理器发送一个API-INTERROGATE-MSG信息。

询问信息的使用使得演示程序的使用有很大的灵活性。由于采用一个询问功能，演示程序就能定位一个目标程序，确定那些目标程序处于特定位置上，并确定一个目标程序的状态。这些询问功能的采用可使演示程序执行这样一项演示，即使该演示程序开始并不知道在演示中使用的目标程序的位置、标识符和／或状态。

如果用户想打开档案文件Fred，而不是接收一个演示，则过程“Process-open”的51～64行的指令将使代理器108在执行由“NewWave Office”应用程序100产生的命令之前先监

视这些命令。用户所产生的一个命令，当被从动作处理器１０１送至命令处理器１０２时被截听。

第５１～６４行中的指令使用户能以多种方式打开档案文件Fred。例如，用户可以把箭头３０３置于档案文件Fred处并二次按压按键２７，或者，用户可以选择档案文件Fred并从一个下移的菜单中选用命令“Open”。不管用户选用何种打开档案文件Fred的方法，两个命令必须依次发生。首先，在用户已选用档案文件Fred的情况下，“选择”命令必须回送到命令处理器１０２。当发生这种情况时，在第５３行中的Class# 被赋以值“FOLDER”（档案文件），第５４行中的Title# 被赋以值“Fred”，此“选择”命令在第５５行执行。其次，一个“Open”命令必须回送到命令处理器１０２供执行。当一个“Open”命令被回送时，并且，如果Class# = Folder 和 Title# = Fred，则第５８行中的Open-flag#将被置“１”，此Open命令将在第５９行执行。

所有其它的命令被代理器１０８所忽略。当用户成功地打开档案文件Fred时，窗口３１９出现，如图１７所示。

在第２５行，监视被关闭。第２６和２７行的指令被代理器108送至目标程序“Lesson Instruction"("课程说明”）供执行。执行这些指令时，窗口３１１从显示器１４移去，窗口３１６出现，如图１７所示。在第２８行的指令重新定义监视过程为过程“Process-button”，使得在用户执行任何命令时，过程“Process-button”被执行，但是仅当监视被接通时才如此。

在第２９行，变量“button-flag#”被置“０”。在第３０行，指令“Set Command on”（接通命令）使得监视接通，以

便在任何一个应用程序中产生一个命令时也产生一个命令陷阱。

只要button-flag# 保持为0，代理器108就循环执行第31～33行之间的指令。在第32行指令处，代理器108等待从任何应用程序截听来的一个命令。当一个来自INSTRUCTION 应用程序200的命令被截听时，button-flag# 被置为“0”，代理器108退出循环，继续执行第34行的指令。

在第34行，监视被关闭。第35行的指令被代理器108送至目标程序“Lesson Instruction"("课程说明”)执行。执行此指令时，窗口316从显示器14移去。

附录A含有API130的介绍(程序员使用指南第4章)。

附录B为开发代理任务语言的准则(代理任务语言准则)。

附录C含有任务语言详细说明。

附录D含有API-INTERROGATE-MSG介绍。

附录E为一篇题为“可扩展的代理任务语言”的论文。

附录F含有HP公司的内部文件，题为“在NewWave上实施以计算机为基础的培训的工具和策略”。

附录G含有一篇刊登在HP公司杂志上的文章，题为“以计算机为基础的培训设施(“Cousteau”)”。

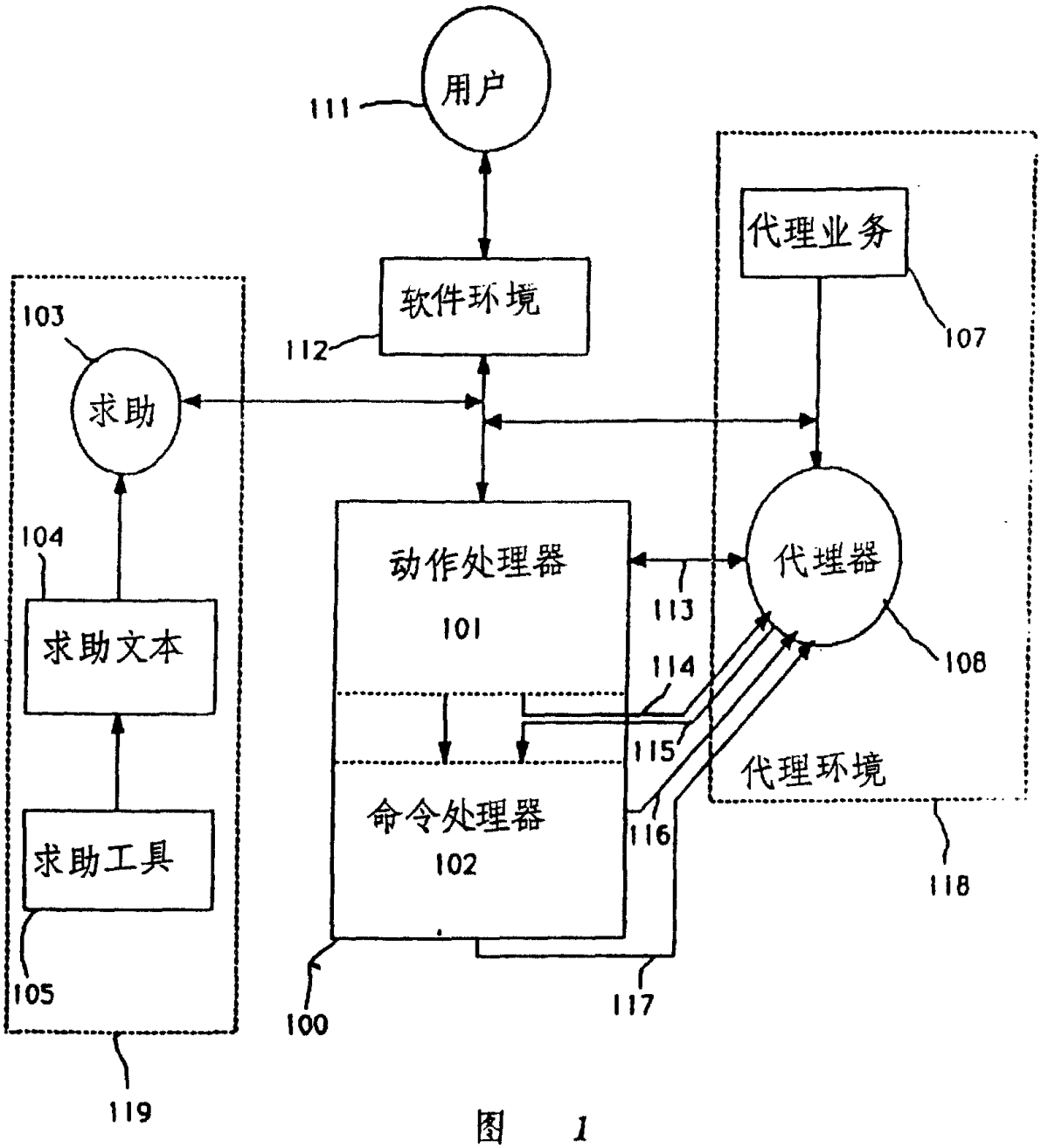

用户
111
软件环境
112
代理业务
107
103
求助
104
求助文本
求助工具
105
119
动作处理器
101
命令处理器
102
100
113
114
115
116
117
代理器
108
代理环境
118

图　1

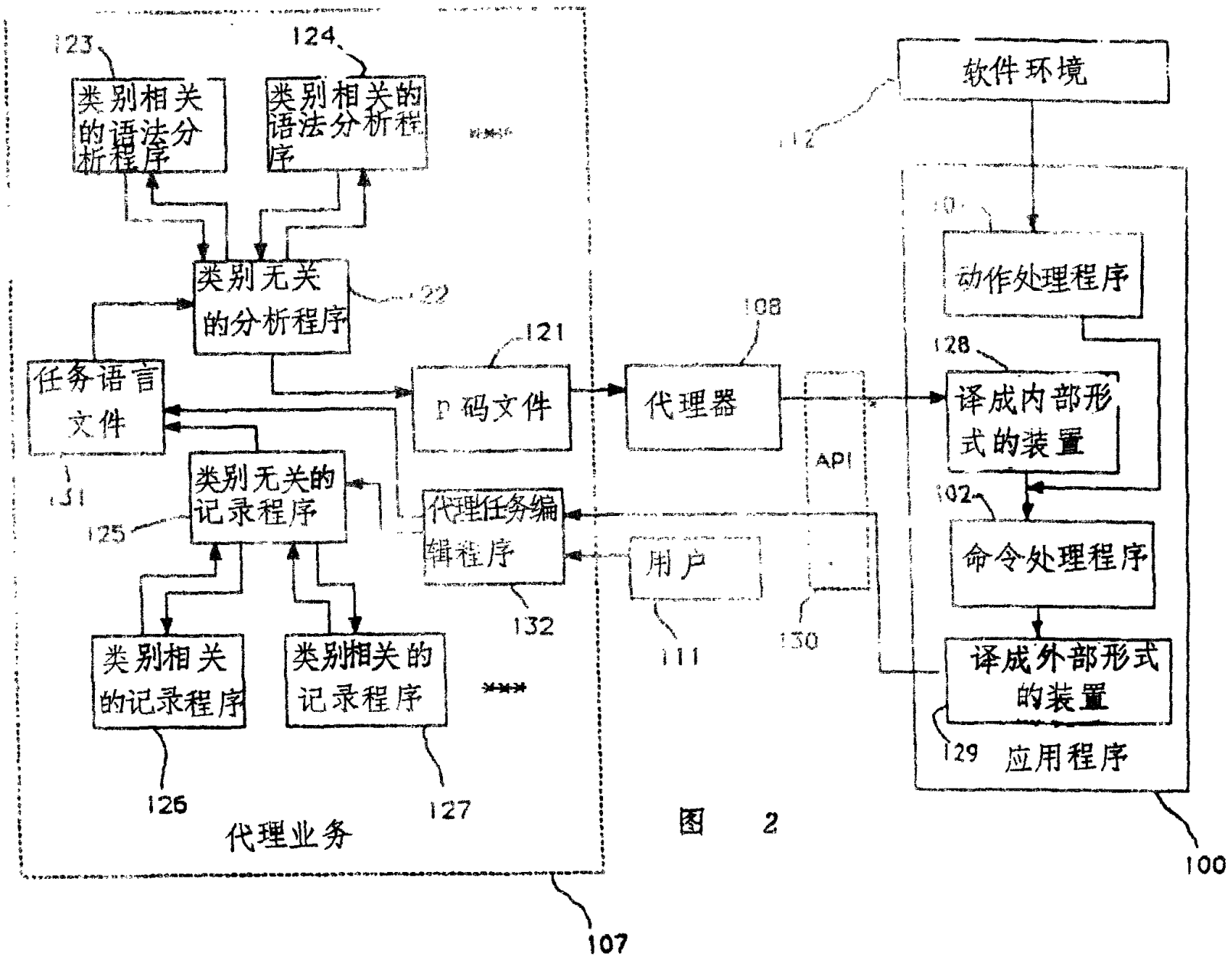

123
124
类别相关的语法分析程序
类别相关的语法分析程序
类别无关的分析程序
122
任务语言文件
131
121
P码文件
108
代理器
API
软件环境
112
110
动作处理程序
128
译成内部形式的装置
102
命令处理程序
译成外部形式的装置
129
应用程序
100
类别无关的记录程序
125
代理任务编辑程序
132
用户
111
130
类别相关的记录程序
126
类别相关的记录程序
127
代理业务
107

图 2

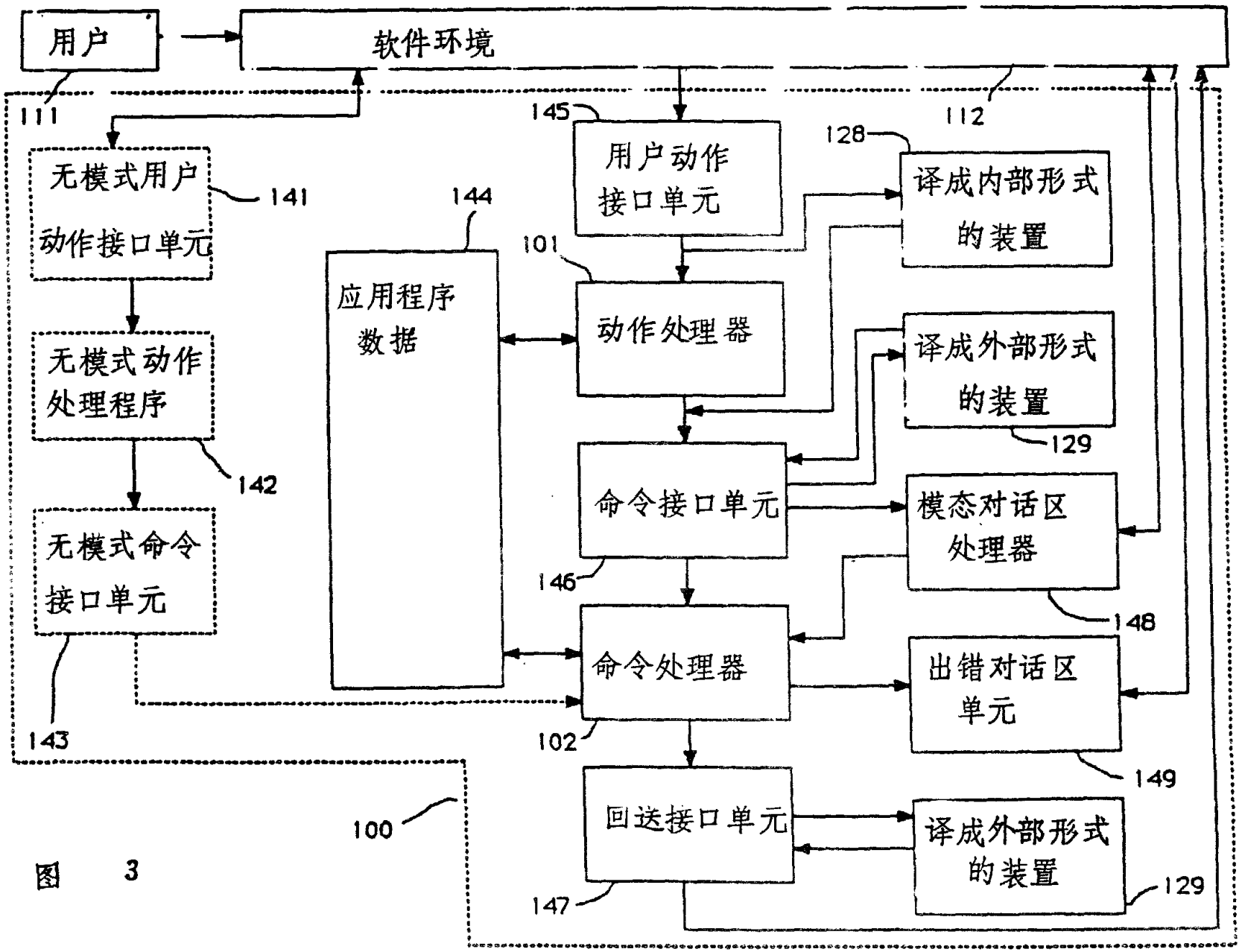

用户
软件环境
111
112
145
用户动作
接口单元
128
译成内部形式
的装置
无模式用户
动作接口单元
141
144
101
应用程序
数据
动作处理器
译成外部形式
的装置
129
无模式动作
处理程序
142
命令接口单元
模态对话区
处理器
146
148
无模式命令
接口单元
命令处理器
出错对话区
单元
143
102
149
100
回送接口单元
译成外部形式
的装置
147
129

图 3

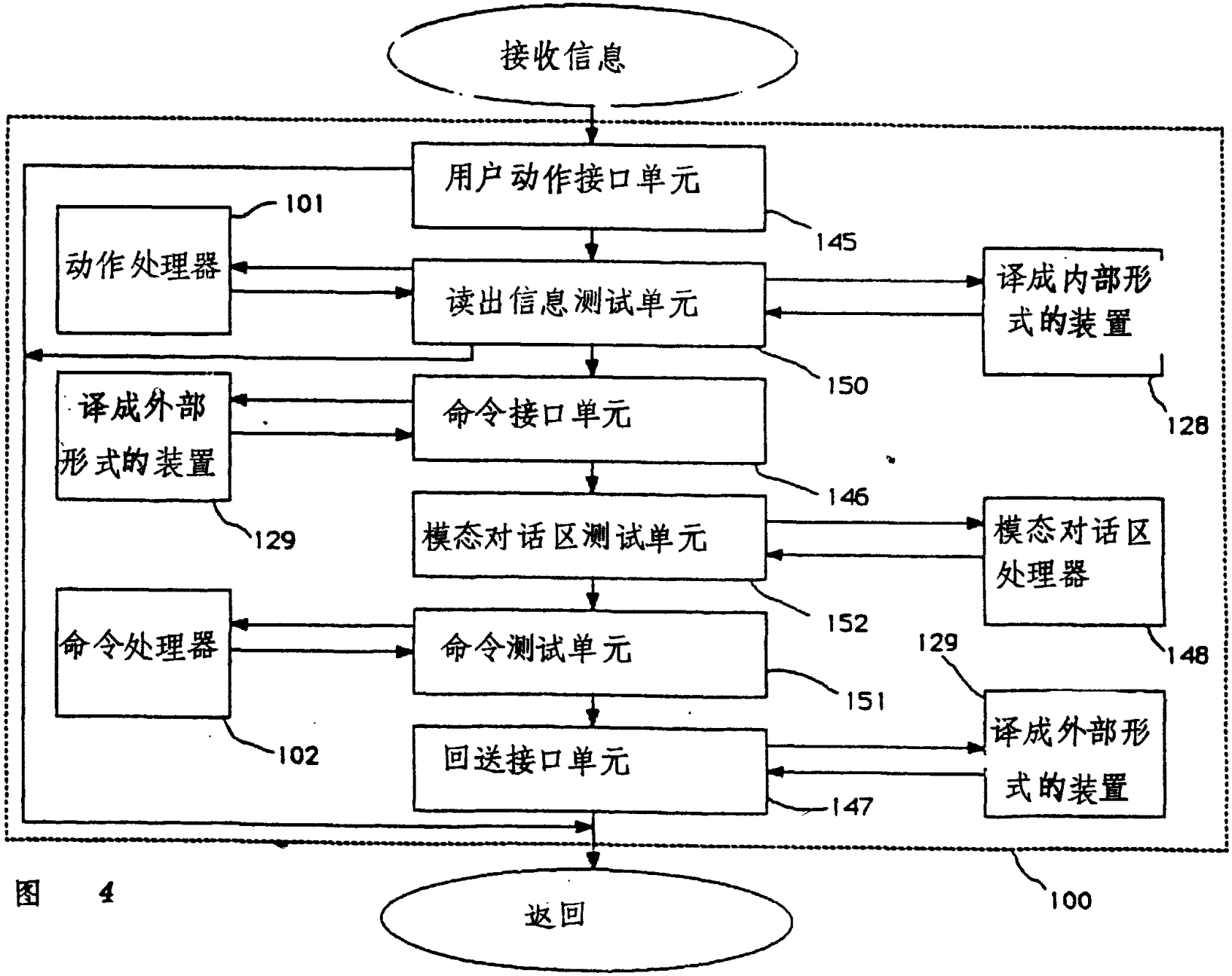

接收信息
用户动作接口单元
145
101
动作处理器
读出信息测试单元
译成内部形式的装置
150
128
译成外部形式的装置
命令接口单元
129
146
模态对话区测试单元
模态对话区处理器
152
148
命令处理器
命令测试单元
129
151
102
回送接口单元
译成外部形式的装置
147
100
返回

图 4

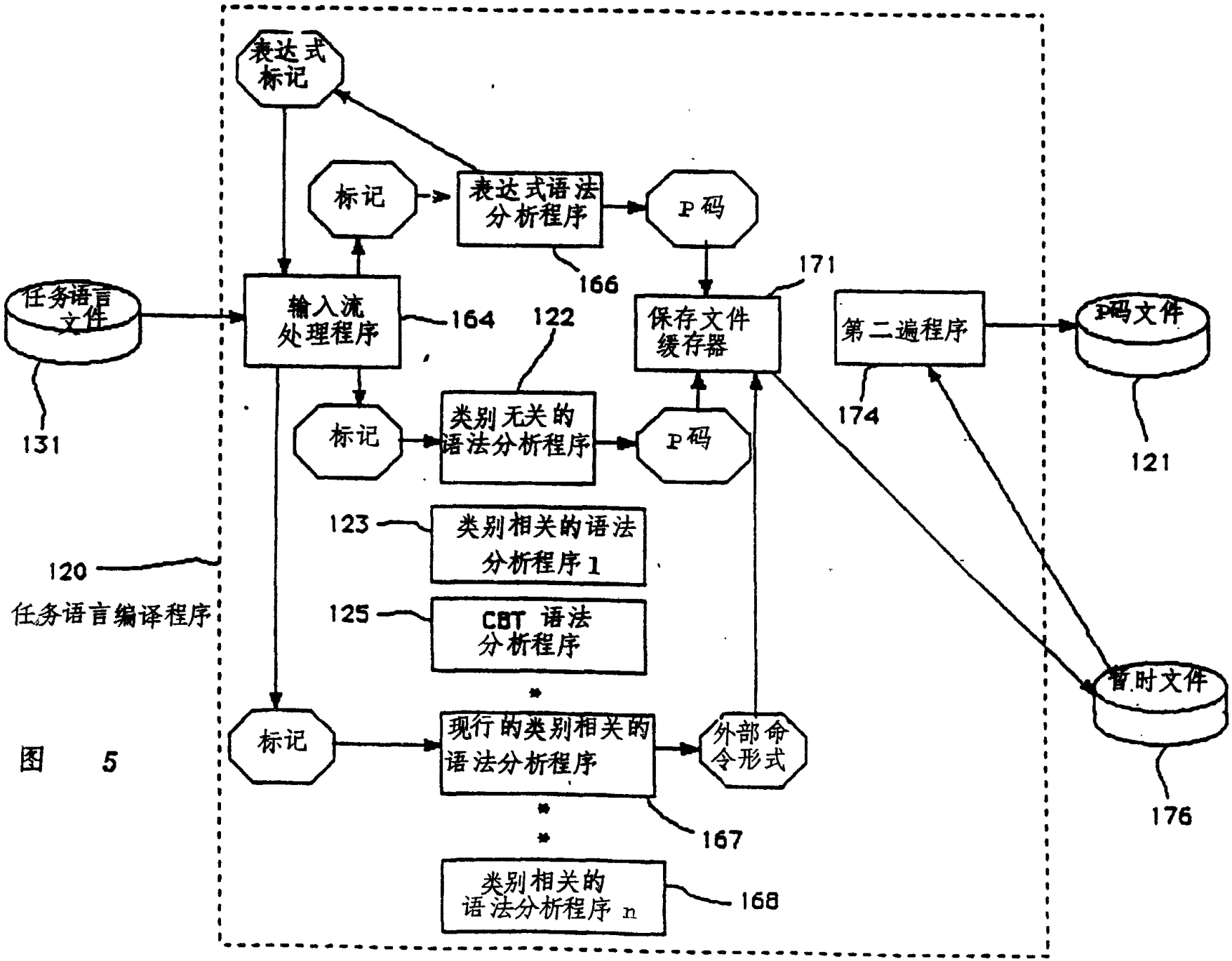

表达式标记
标记
表达式语法分析程序
P码
166
171
任务语言文件
输入流处理程序
164
122
保存文件缓存器
第二遍程序
P码文件
131
标记
类别无关的语法分析程序
P码
174
121
123
类别相关的语法分析程序1
120
任务语言编译程序
125
CBT 语法分析程序
暂时文件
标记
现行的类别相关的语法分析程序
外部命令形式
167
176
类别相关的语法分析程序 n
168

图 5

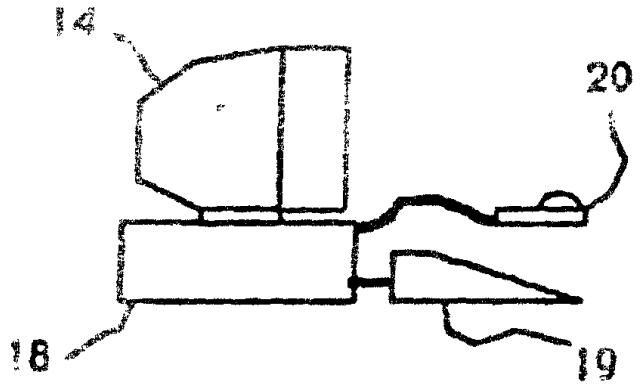
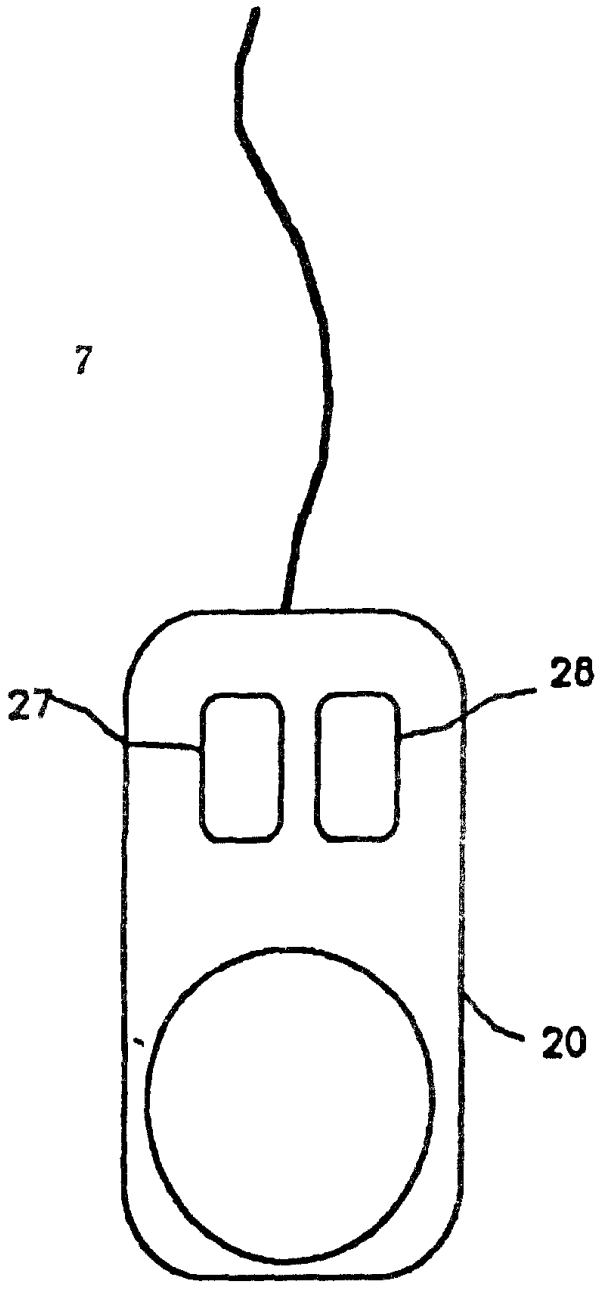


図　6

图 7



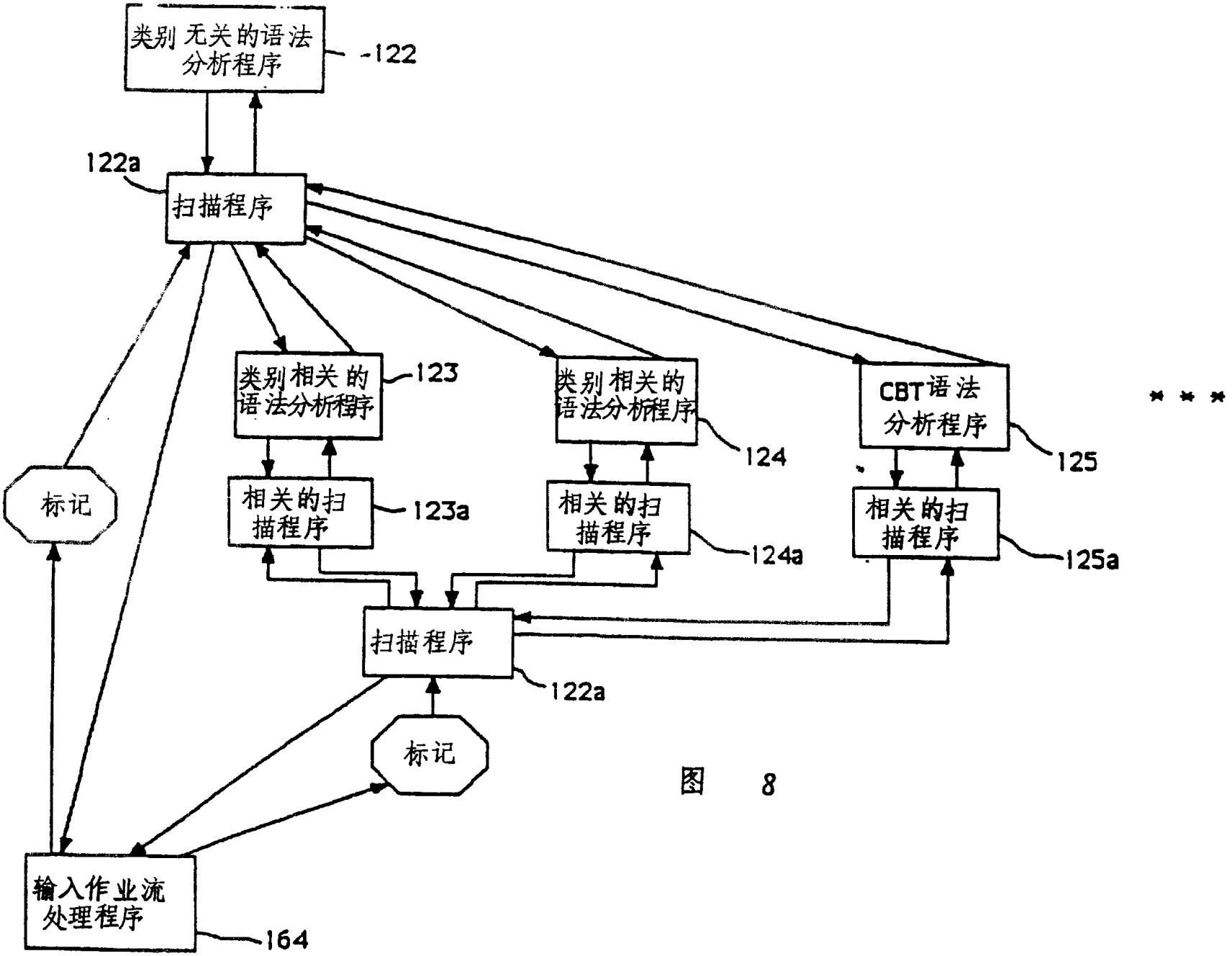

类别无关的语法分析程序
122
122a
扫描程序
类别相关的语法分析程序
123
类别相关的语法分析程序
124
CBT语法分析程序
125
标记
相关的扫描程序
123a
相关的扫描程序
124a
相关的扫描程序
125a
扫描程序
122a
标记
输入作业流处理程序
164

图 8

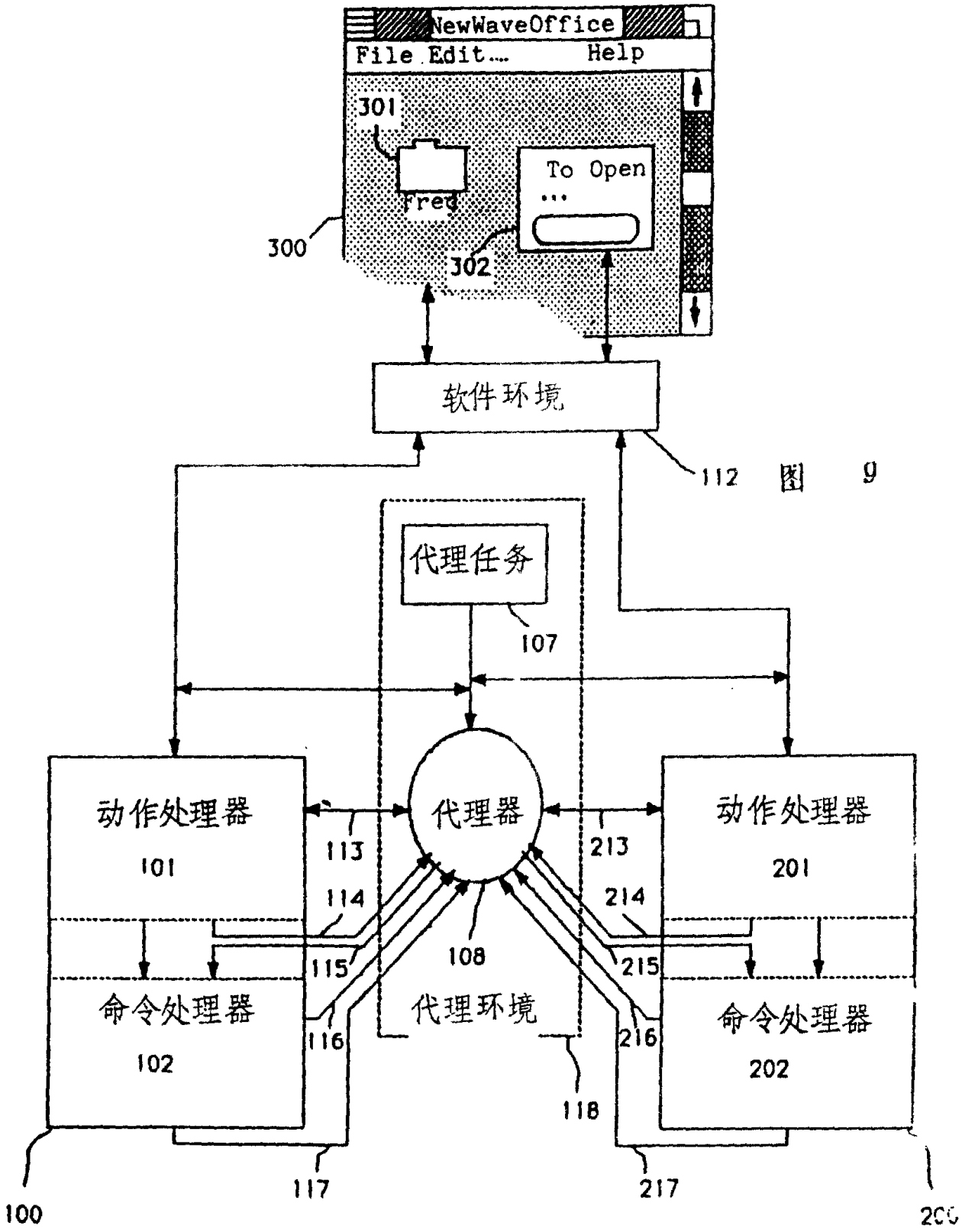

NewWaveOffice
File Edit.... Help
301
To Open
...
300
302
软件环境
112
代理任务
107
动作处理器
101
代理器
动作处理器
201
113
213
114
214
115
108
215
命令处理器
102
代理环境
116
216
命令处理器
202
118
117
217
100

图 9

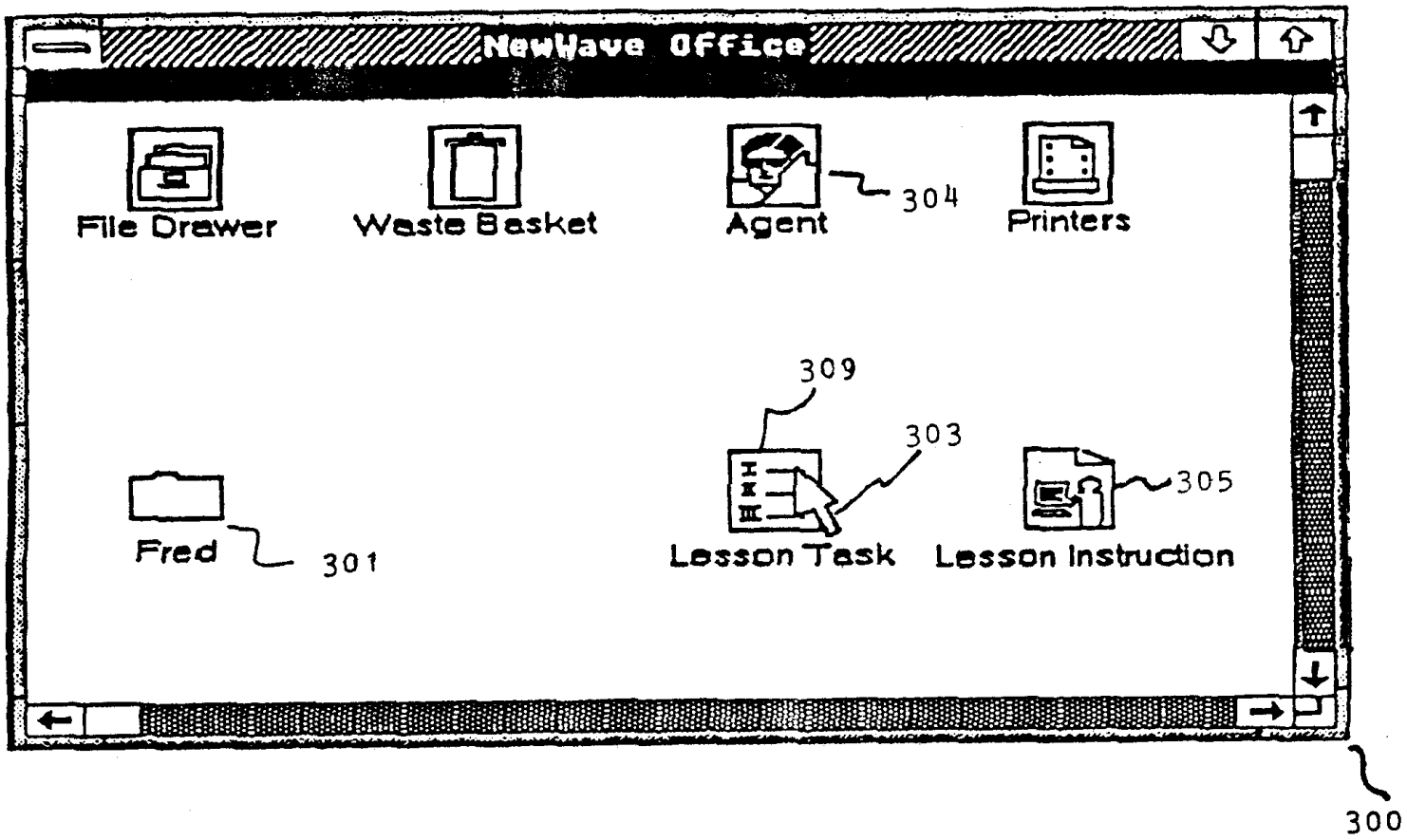

NewWave Office
File Drawer
Waste Basket
Agent
304
Printers
309
303
305
Fred
301
Lesson Task
Lesson Instruction
300

图 10

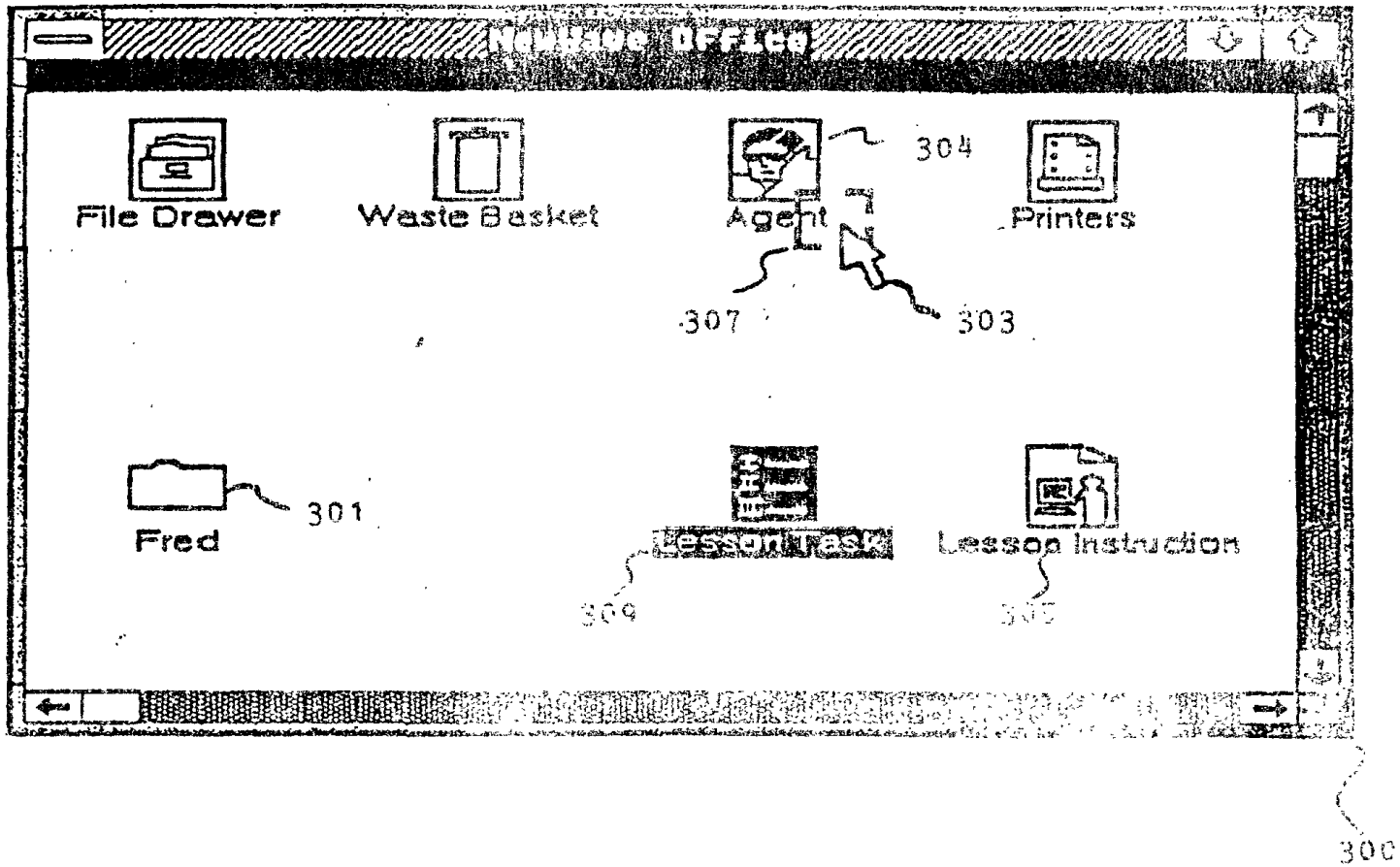

File Drawer
Waste Basket
Agent
304
Printers
307
303
301
Fred
Lesson Task
Lesson Instruction
309
305
300

图 1 1

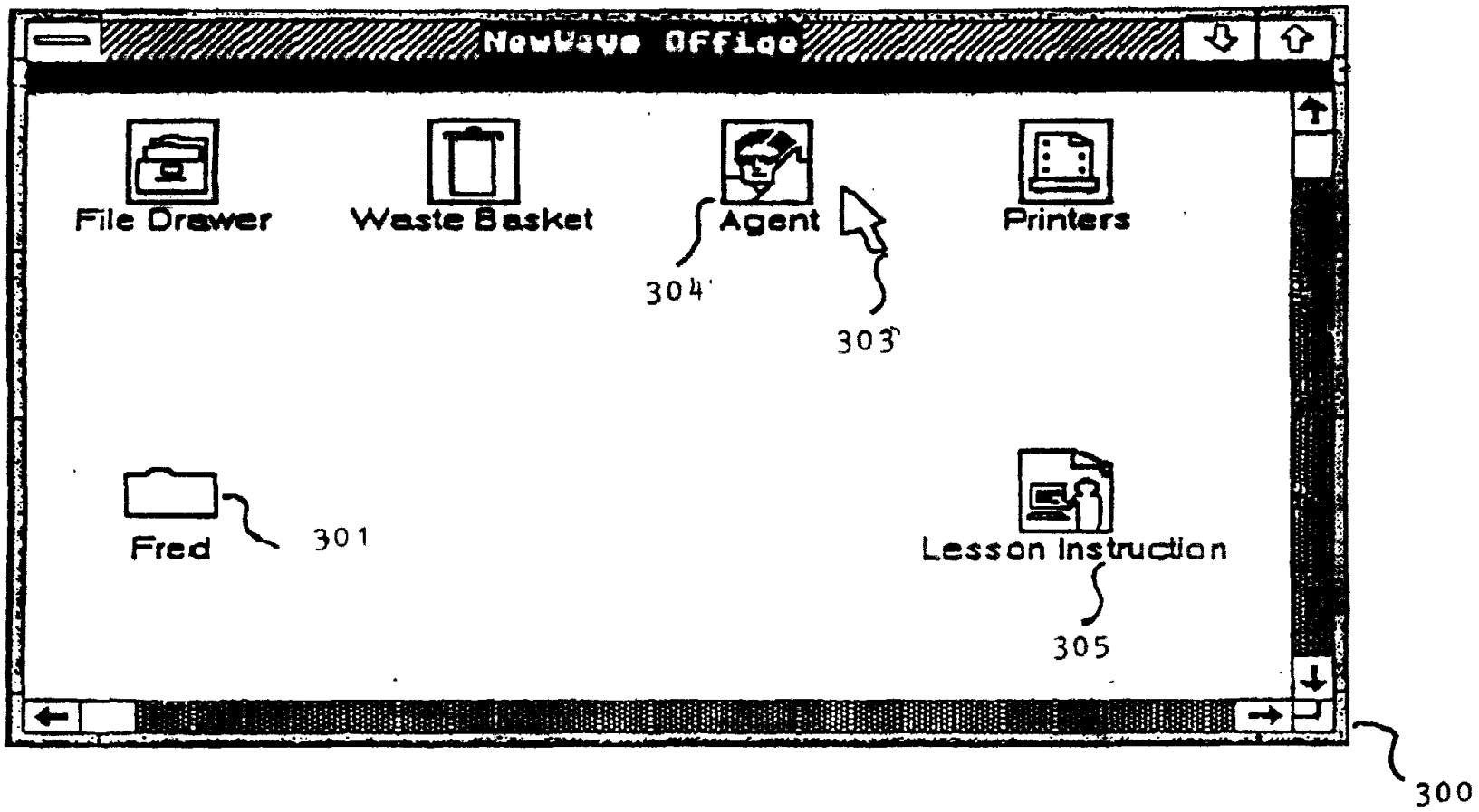

NewWave Office
File Drawer
Waste Basket
Agent
Printers
304
303
Fred
301
Lesson Instruction
305
300

图 1 2

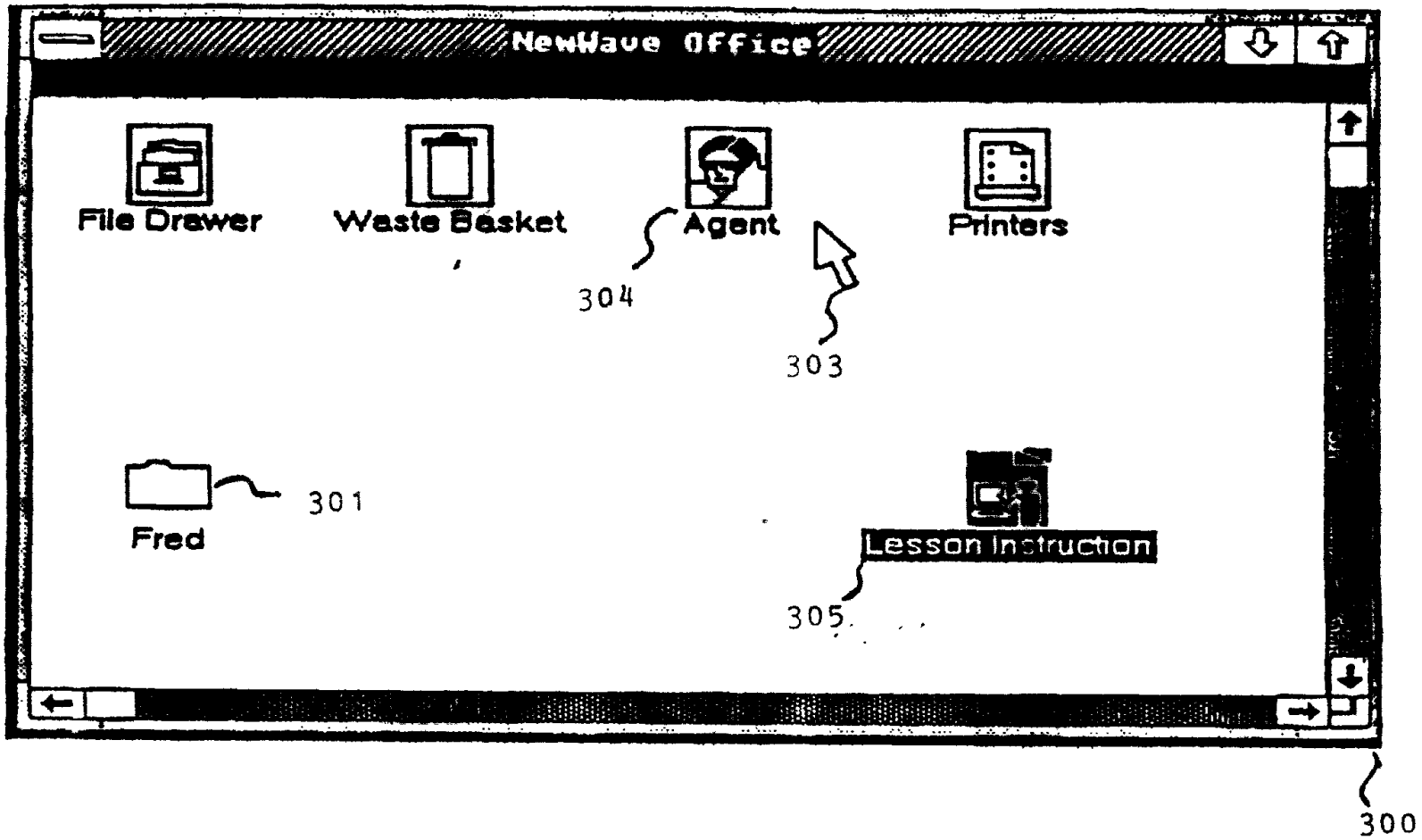

NewWave Office
File Drawer
Waste Basket
Agent
Printers
304
303
301
Fred
Lesson Instruction
305
300

图 13

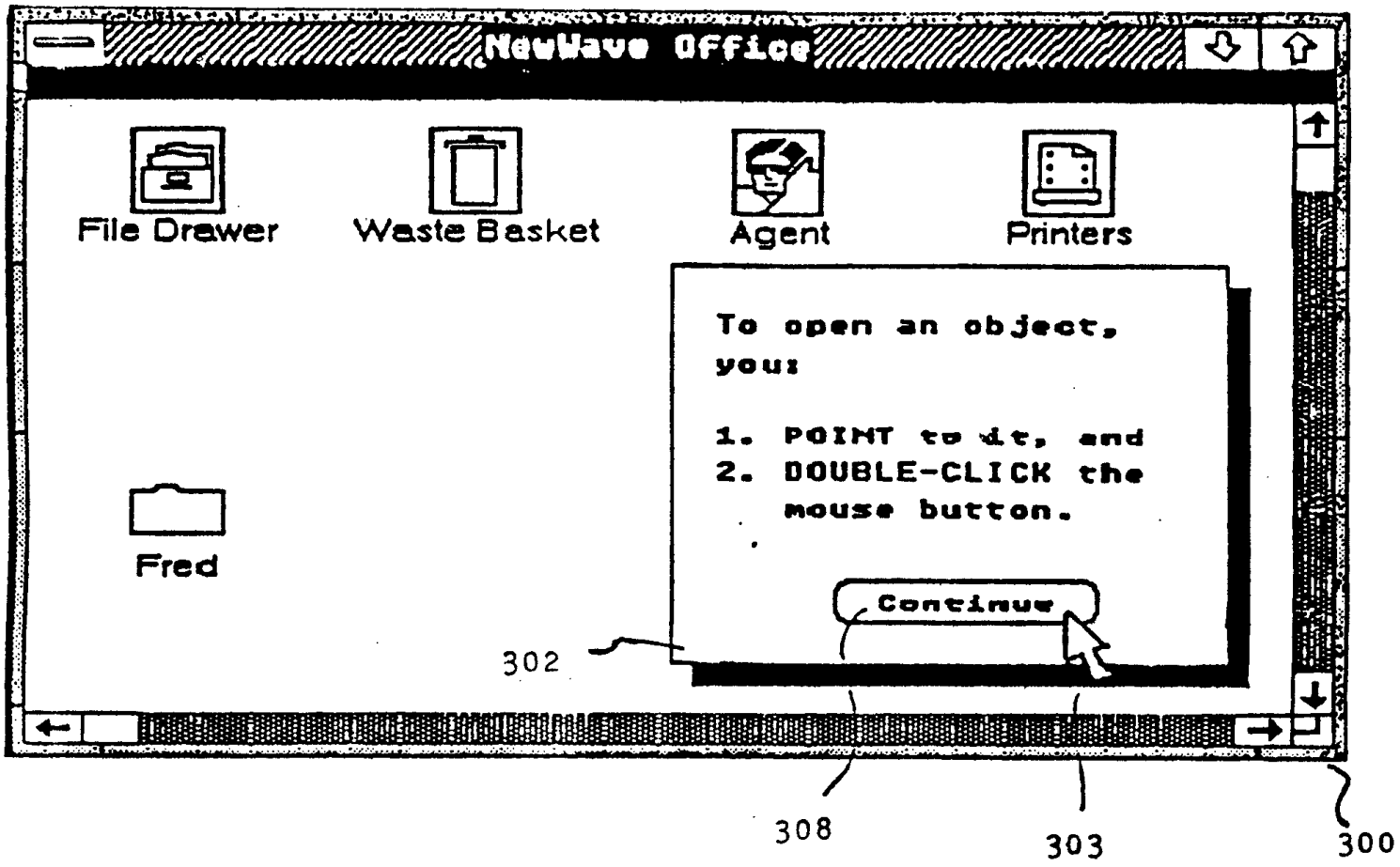

NewWave Office
File Drawer
Waste Basket
Agent
Printers
To open an object, you:
1. POINT to it, and
2. DOUBLE-CLICK the mouse button.
Continue
Fred
302
308
303
300

图 14

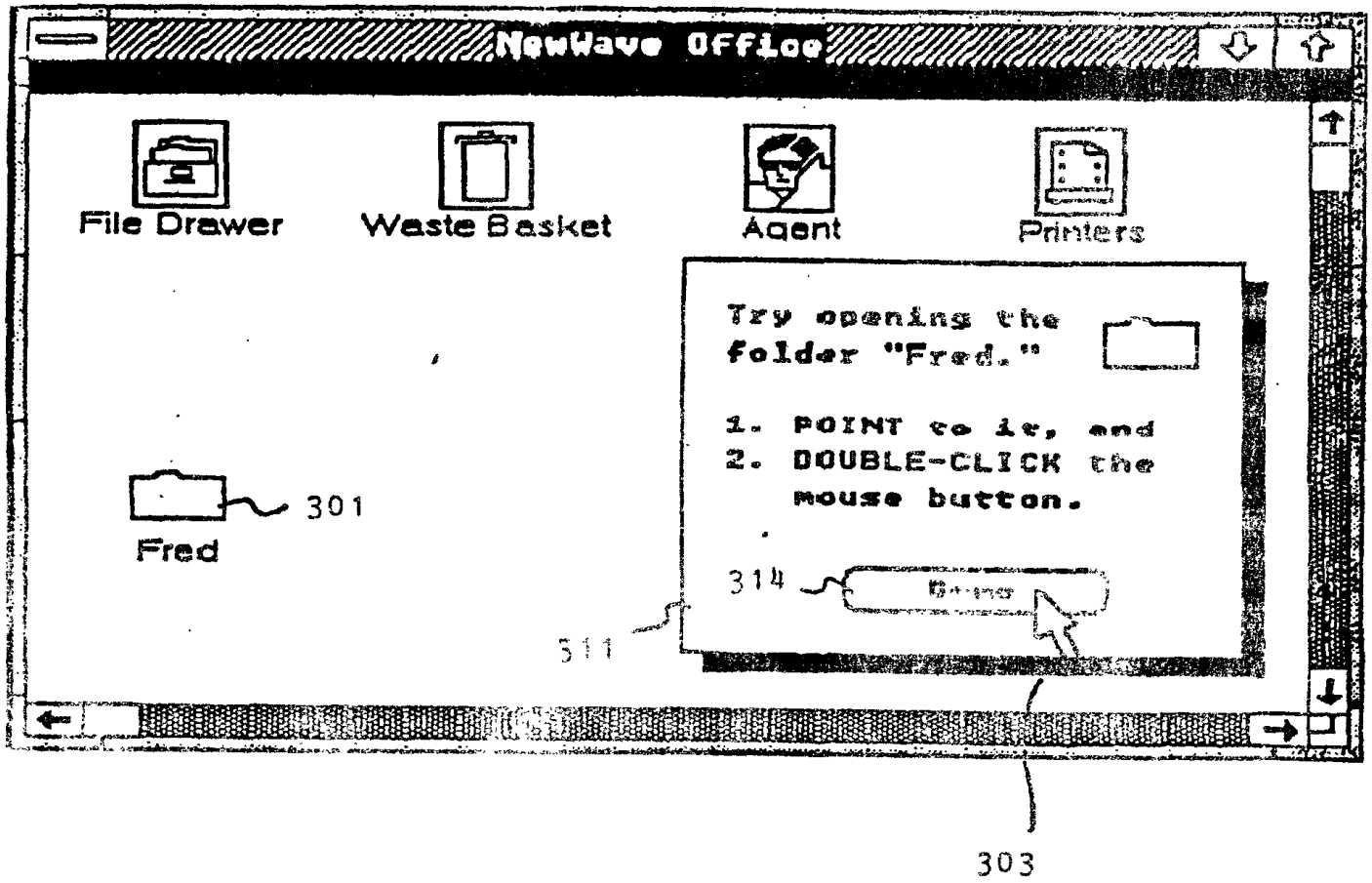

NewWave Office
File Drawer
Waste Basket
Agent
Printers
Try opening the folder "Fred."
1. POINT to it, and
2. DOUBLE-CLICK the mouse button.
301
Fred
314
511
303

図 5

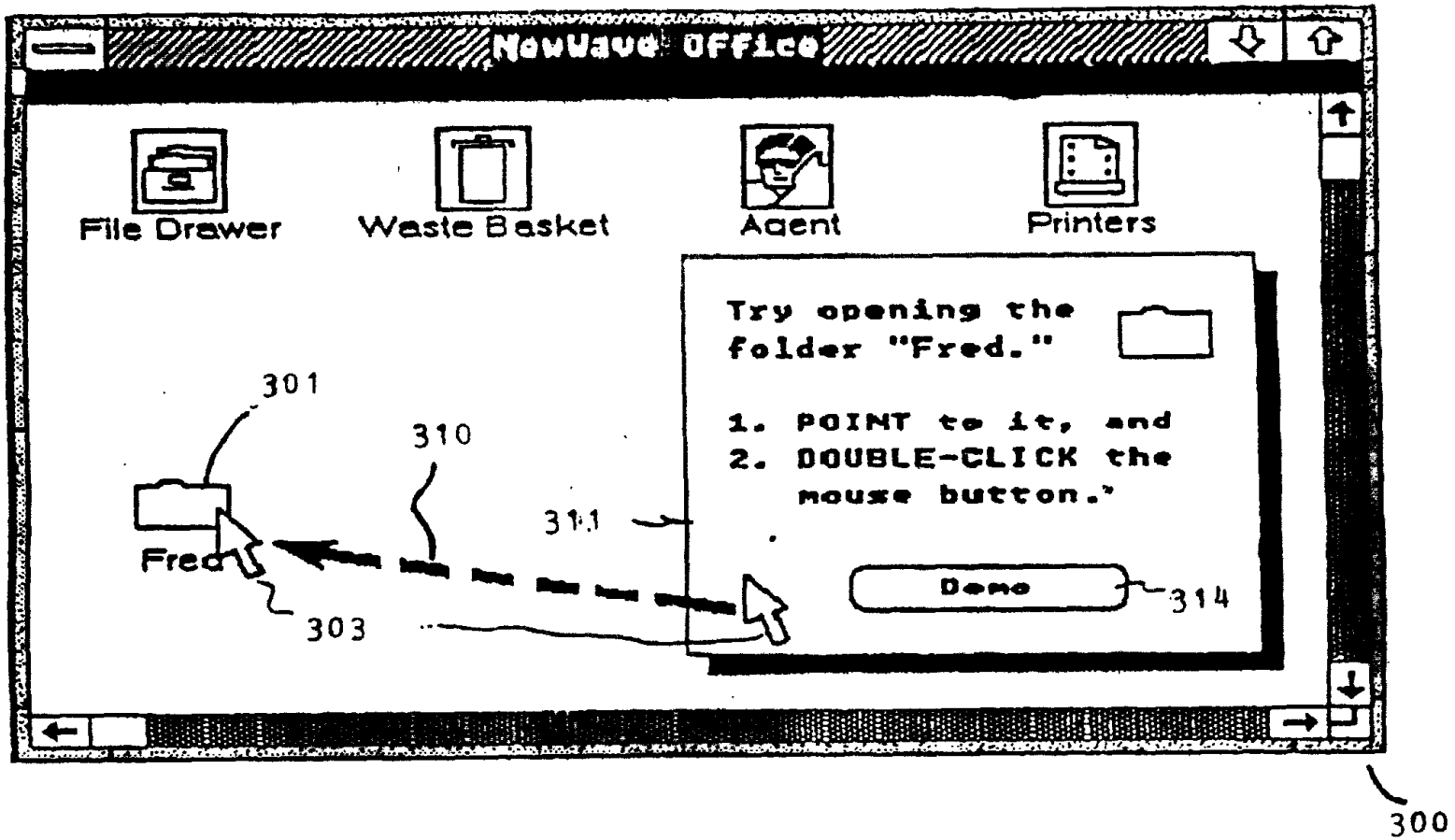

NewWave Office
File Drawer
Waste Basket
Agent
Printers
Try opening the folder "Fred."
1. POINT to it, and
2. DOUBLE-CLICK the mouse button."
Demo
Fred
301
310
311
303
314
300

图 16

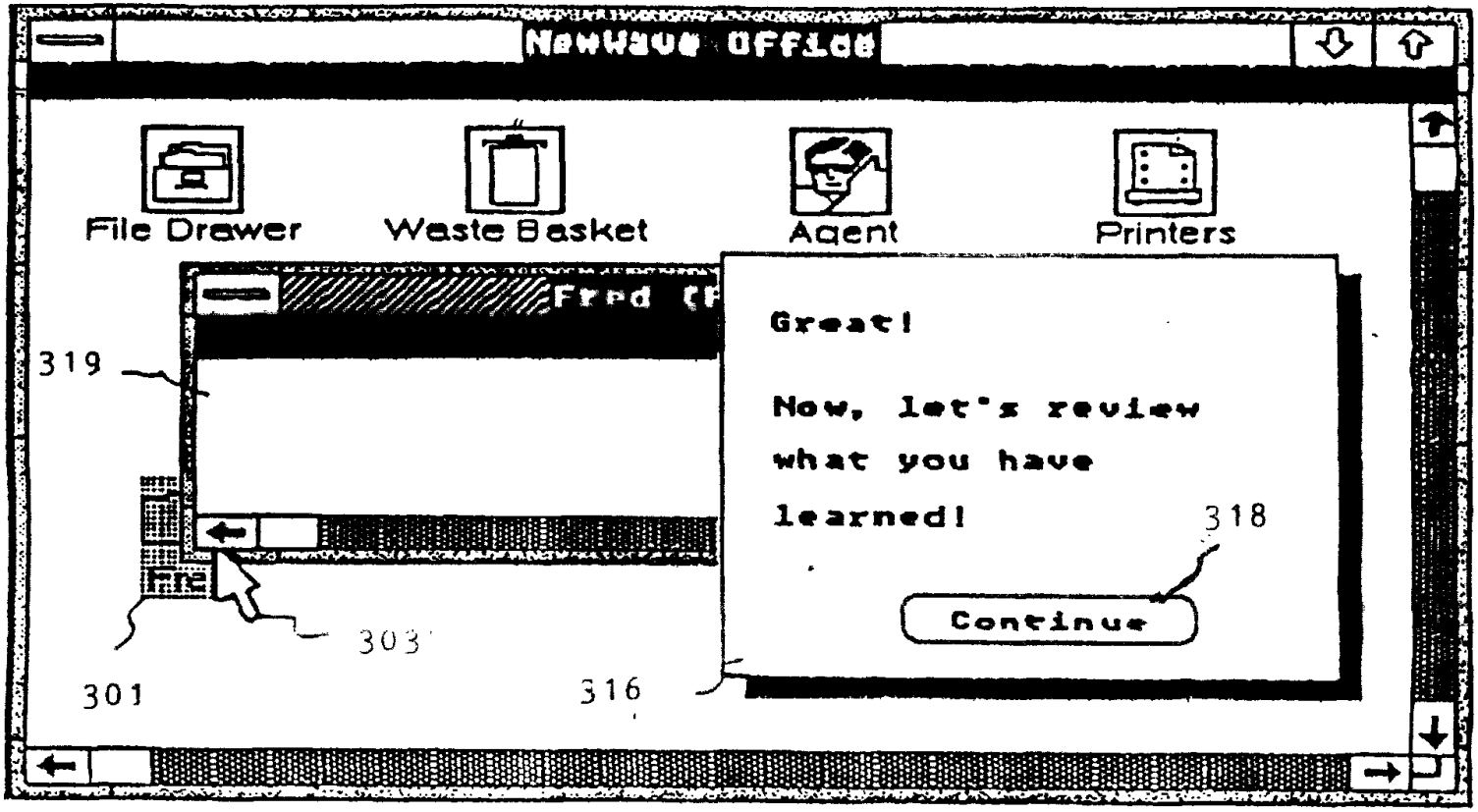

File Drawer
Waste Basket
Agent
Printers
319
Great!
Now, let's review what you have learned!
318
Continue
303
301
316

图 17