[19]中华人民共和国专利局

[11] 公开号 CN 1116007A

[12] 发明专利申请公开说明书

[21]申请号 94190839.9

[51]Int.Cl[6]

G06F 9/44

[43]公开日 1996 年 1 月 31 日

[22]申请日 94.1.3

[30]优先权

[32]93.8.25 [33]US[31]08/112,821

[86]国际申请 PCT/US94/00012 94.1.3

[87]国际公布 WO95/06282 英 95.3.2

[85]进入国家阶段日期 95.6.27

[71]申请人 塔里根特公司

地址 美国加里福尼亚州

[72]发明人 福兰克·T·恩固因

戴维·R·安德逊

蒂莫西·J·O·卡特林

[74]专利代理机构 北京市中原信达知识产权代理公司

代理人 余 朦 余 刚

权利要求书 5 页 说明书 21 页 附图页数 7 页

[54]发明名称 面向目标的导航系统

[57]摘要

用于在面向目标的计算机系统中进行导航的技术。该计算机系统包括具有存储器和显示器的处理器。第一文件被装入存储器并在显示器上表示。然后，生成一个选择目标并将其与显示上的第一文件关联。引用与存储器中的第一选择关联。然后，第二文件被装入存储器并表示在显示器上，且一个引用与第二文件中对第一选择关联。最后，通过在第二文件中对第一选择的引用而启动导航。亦可在目标位于本地的情况下通过在一个远地目标上的引用执行命令。

权利要求书

1. 一种用于计算机系统把一个第一目标导航到一个第二目标的方法，该计算机系统包括一个处理器，一个与处理器相连且由处理器控制的存储器，一个处理器相连并由处理器控制的显示器，该方法的特征在于包括：以下步骤：

(a)将第一文件装入存储器并在显示器上表示该第一文件；

(b)在存储器中生成与一个第一选择关联的选择目标；

(c)在存储器中生成一个与第一选择关联的引用；

(d)把一个第二文件装入存储器并在显示器上表示该第二文件；

(e)把与第一选择关联的引用放入第二文件；以及

(f)通过在第二文件中引用第一文件中的第一选择而进行导航。

2. 权利要求 1 所述方法，包括通过在第二文件中引用第一文件中的第一选择从而进行导航，从而在显示器上表示当前更新的步骤。

3. 权利要求 1 所述方法，还包括通过在第二文件中引用第一文件中的第一选择而进行导航，从而在一个网络中存取信息的步骤。

4. 权利要求 1 所述方法，还包括在目标位于本地系统情况下在一个远地目标上执行命令的步骤。

5. 权利要求 1 所述方法，还包括通过在第二文件中引用第一文件的相关部分进行导航从而显示第一文件相关部分的步骤。

6. 权利要求 1 所述方法，其中的第一和第二文件是电子数据表。

7. 权利要求 6 所述方法，还包括通过在第二电子数据表中引用第一电子数据表的第一选择而进行导航以及把该方法或用于一个第一电子数据表的信息，而把一个方程式用于显示器上的第一电子数据表的信息。

8. 权利要求 1 所述方法，还包括把对于一个第二折合库的引用放在一个第一折合库之内以便在第一折合库打开时启动从第一折合库到第二折合库的导航的步骤。

9. 权利要求 1 所述方法，还包括以下步骤：

(a)生成计算机资源的引用；

(b)在存储器中存储该资源；

(c)在显示器上显示该资源；

(d)响应一个光标信号在该引用附近的任何图符上执行图符动作。

10. 权利要求 9 所述方法，还包括把一个文件拖移到该引用并把该文件投入到该引用上以激发图符动作的步骤。

11. 权利要求 9 所述方法，还包括作为该图符动作而删除一个文件的步骤。

12. 权利要求 9 所述方法，还包括作为该图符动作而复制一个文件的步骤。

13. 权利要求 9 所述方法，还包括作为该图符动作而打印一个文件的步骤。

14. 权利要求 9 所述方法，还包括作为图符动作而传真一个文件的步骤。

15. 权利要求 9 所述方法，还包括作为图符动作而启动一个电话通讯的步骤。

16. 权利要求 9 所述方法，还包括作为图符动作而存储一个文件的步骤。

17. 一种用于计算机系统把一个第一目标导航到一个第二目标的装置，该计算机系统包括一个处理器，一个与处理器相连且由处理器控制的存储器，一个与处理器相连并由处理器控制的显示器该装置的特征在于包括：

(a)用于将第一文件装入存储器并在显示器上表示该第一文件的装置。

(b)在存储器中生成与一个第一选择关联的选择目标的装置；

(c)在存储器中生成一个与第一选择关联的引用的装置；

(d)把一个第二文件装入存储器并在显示器上表示该第二文件的装置。

(e)把与第一选择关联的引用放入第二文件中的装置，以及

(f)通过在第二文件中引用导航的装置第一文件中的第一选择而进行导航的装置。

18. 权利要求 17 所述装置，包括通过第二文件中引用第一文件中的第一选择而进行导航从而在显示器上表示当前更新的装置。

19. 权利要求 17 所述装置，还包括通过在第二文件而引用第一文件中的第一选择进行导航，从而在一个网络中存取信息的装置。

20. 权利要求 17 所述装置，还包括在目标位于本地系统情况下在一个远地目标上执行命令的装置。

21. 权利要求 17 所述装置，还包括通过在第二文件中引用第一文件的相关部分进行导航而显示第一文件相关部分的装置。

22. 权利要求 17 所述装置，其中的第一和第二文件是电子数据表。

23. 权利要求 22 所述装置，还包括通过在第二电子数据表中引用第一电子数据表的第一选择而进行导航以及把该方法或用于一个第一电子数据表的信息，而把一个方程式用于显示器上的第一电子数据表的信息的装置。

24. 权利要求 17 所述装置，还包括把对于一个第二折合库的引用放在一个第一折合库之内以便在第一折合库打开时启动从第一折合库到第二折合库的导航的装置。

25. 权利要求 7 所述装置，还包括：

(a)用于生成对计算机资源的引用的装置；

(b)用于在存储器中存储该资源的装置；

(c)用于在显示器上显示该资源的装置；以及

(d)用于响应一个光标信号在该引用时近的任何图符上执行图符动作的装置。

26. 权利要求 25 所述方法，还包括把一个文件拖移到该引用并把该文件投入到该引用上以激发图符动作的装置。

27. 权利要求 25 所述装置，还包括由该图符动作而删除一个文件的装置。

28. 权利要求 25 所述装置，还包括由该图符动作而复制一个文件的装置。

29. 权利要求 25 所述装置，还包括由该图符动作而打印一个文件的装置。

30. 权利要求 25 所述装置，还包括作为图符动作而传真一个文件的装置。

31. 权利要求 25 所述装置，还包括作为图符动作而启动一个电话通讯的装置。

32. 权利要求 25 所述方法，还包括作为图符动作而存储一个文件的装置。

说　明　书

面向目标的导航系统

本发明涉及计算机系统的改进，尤其涉及在系统和信息区域之间作为整体进行导航的系统和方法，并涉及在本地位置的情况下使用远地目标。

本申请的部分内容受版权保护。版权持有人允许该相关部分作为专利文件而被复制和公开，但保留其他权利。

本申请涉及由 Debra L. Orton 等在 1992 年 12 月 23 日提交给美国联邦专利商标局并转让给本申请人的题为“面向目标的结构系统”的专利申请。该申请亦于 1993 年 12 月 23 日提交到中国专利局。其内容由本申请所引用。

本申请还涉及由 Arn Schaefer 等人于 1993 年 7 月 9 日提交到美国联邦专利商标局并转让给本申请人的题为“动态连接系统”的专利申请，该申请亦于 1994 年 11 月 3 日提交了 PCT 申请，其 PCT 申请号为 PCT/US94/00099，该申请并于 1995 年 1 月 19 日提交到中国专利局，其申请号为 94 903087。该申请的内容亦被本申请所引用。

本申请还涉及由 Frank Nguyen 于 1993 年 6 月 3 日提交到美国联邦专利商标局并转让给本申请人的题为“储存器目标系统”的专利申请，该申请于 1994 年 1 月 10 日提交了 PCT 申请，申请号为 PCT/US94100269，该申请亦于 1995

年4月3日提交到中国专利局，其申请号为94190576.4。该申请的内容亦为本申请所引用。

计算机系统通常需要在应用程序(例如在用于控制炼油的处理控制系统中的程序)之间进行连接。因此，在图形工作站处的操作员可以选择指示一个阀的图形符号，并将其状态从关闭改变为开启以便实现到其在显示上的图象的改变和到该阀门被关闭的处理环境的相应改变。但是，为完成特定任务大部分处理控制系统是"硬编码(hard—code)"。因此不易通过操作员的控制台对连接进行再定义。进而，这些连接不灵活并且不是多目的的，除非每个连接都有一个可以容易改变的预定义的任务。

最近，苹果和其他交互式计算机系统在关闭的文件和储存器或折合库之间提供了连接。因此，例如一个交互计算机系统的用户可以通过图符选择指示文件和打开文件的图符。一个折合库可以有一个通过表示该折合库的图符而投入其中的选定文件。

本发明通过提供在计算机系统中的目标之间进行导航的系统和方法，并把计算机中的远地目标用作本地目标而克服了先有技术的不足。在面向目标的计算机系统中的导航包括带有存储器和显示器的处理器。然后，生成选择的目标并将其与显示器上的第一文件关联。一个引用与存储器中的第一选择相关联。其后，将第二文件装入存储器并在显示器上展示，并把一个引用与第二文件的第一选择关联。最后，通过在第二文件中对第一文件的第一选择的引用而允许导航。

图 1 是根据本发明一个最佳实施例的个人计算机框图；

图 2 是根据本发明具有选择区域和引用目标的文件显示展示；

图 3 是根据本发明的最佳实施例生成一个引用目标的处理流程图；

图 4 是根据本发明最佳实施例使用一个引用目标的逻辑流程图；

图 5 是根据一个最佳实施例具有对一个远地打印机的引用的显示图；

图 6 是根据本发明一个最佳实施例展示用户把一个目标的复制拖至一个引用的示图；以及

图 7 示出了根据本发明一个最佳实施例在部分完成一个动作中与一个引用目标相邻的目标。

导航连接的综述

一个引用目标(reference object)具有区别于图形显示上的其他目标的固有特性。首先，一个引用(reference)可被放在用户数据可放置的任何地方(即在一个文件中，储存器中，位置中或只是在计算机的桌面上)。其次，一个引用涉及一个真实目标而非另一引用。一个引用只引用一个可以包括多个目标的选择。引用继续引用一个与引用的数量或目标的属性(即命名，图符)的变化和/或在储存层级中的位置无关的一个目标。打开一个引用将打开和表示被引用的目标、复制或删除一个引用并不复制或删除引用的目标。由于一个引用并不保证存在有引用的目标，所以引用是由系统预置单方

向的。

引用的继承特点

引用是一个模型锚(它已在题为“动态连接系统”的专利申请中进行了叙述,该申请被结合于此)一个引用继承了下列特性:

一个引用可以追踪其连接到引用的目标。

一个引用可以高速缓冲存储属性即命名,图符等。

一个引用可作为可视图符表示。

使用实例概要

第一组概要展示如何在一个系统中使用引用。

技术论文

一个作者对另一作者的论文的一部分进行引用。该引用准确指定哪一页和哪一段与其主题相关。然后一个读者可以用该引用把第一个作者的论文导航到第二个作者的论文的相关信息部分。

电子数据表

通常,理想的情况是引用在电子数据表中一个单元的方程式的值,该方程式用来计算另一单元的值。一个用户也可用引用来导航该单元,而无论其是否位于相同或不同的电子数据表中。

折合库

用户可以把引用放到位于另一折合库内的折合库,以便引用好象是在不同的折合库层级。打开该引用折合库就把用户带到起始的折合库地址。

工具设备

若用户需要从系统的不同位置访问一个工具设备(例如打印机,传真电话等),则可生成对该工具设备的引用并将其放在理想位置。引用然后可被用作为辅助导航,或该工具设备可通过引用直接使用而无需导航到其实际位置。

名片

对个人打印机的引用可被放在其电子名片上以便该名片的接收者可把对于第一方的打印机的引用导航。

在以下情况中将同折合库和文件的引用来叙述用户和引用的交互。其他类型的引有的行为是类似的。

从一个折合库中把一个文件引用到相同折合库中:一个用户命令把文件A从折合库X中拖出(在拖出时接口Command键)并将其投入用一折合库。它在折合库X中表示为Document A(对Document A的引用)。

从另一个折合库中把一个文本选择引用到一个文件中:用户命令把Document B中的文本选择拖出并将其投入Folder X。它在Folder X中表示为Document B(对Document A的引用)。

从另一文件中把一个文件引用到一个折合库中:用户命令把Folder X中的Document A拖出并将其投入Docament B。这在Document B中表示为Document A(对Document A的引用)。

删除引用:用户从Document B中拖出Document A并将其投入废物筐(对Trash Can的引用)或直接投入Frash

Can 中。

把一个引用从一个折合库复制(移动)到另一折合库:用户用任选(option)把 Document A 从 Folder X 拖到 Folder X。这在 Folder Y 中示为 Document A′。

把引用从一个文件复制(移动)到一个文件:用户用任选把 Document A 从 Folder X 拖到 Document B,它在 Document B 中示为 Document A′。

把引用从一个文件复制(移动)到一个折合库:用户用任选把 Document A′从 Document B 拖到 Folder X。它在 Document B 中示为 Document A″。

详细的类别体系结构

超连接(Hyperlink)是从源锚到目的锚的双向连接。包含(containables)则是从一个空锚到一个目的锚的潜在双向连接。引用是从一个可能的空锚到一个目的锚的潜在双向连接。每一个都有一个用于表示属性的高速缓冲存储,以及追踪连接到目的锚的唯一协议。用于处理由超连接包含和引用展示的类似特性的设计如下:

• 由 TContainPolicy 提供对超连接,包含和引用的支持。TContainPolicy 的系统预置实施是 TStandardContainPolicy。这些类别是 Container Framework 的一部分,它们还为子类别提供两个抽象模型类别以生成新的数据类型。参见以下详述的"模型类别"一节。在选择侧,由 MSelectionCanContainPolicy 提供支持,该类别的系统预置实施是 TSelectionCanContainPolicy。这些类别亦为 Container-

Framework 的部分。ContainerFramework 也为子类别提供两个抽象选择类别以生成新的选择类型。参见以下详述的“选择类别”一节。

• 超连接，包含和引用的表是一个 TModelAnchor 指针表。超连接(hyperlink)是一个 TModelAnchor。包含和引用来自于通用类型 TContainable，它是 TModelAnchor 的子类别。参见如下详述的“超连接/包含/引用类别”一节。

通过选择目标或目标集以及拖移和投入选择的命令，用户可以生成引用。这一动作把具有源模型选择(即被拖移的目标)的 MakeReference 命令作为一个参数送往源模型，并把一个具有目标模型选择(即整个目标模型)的 Paste 命令作为一个参数送至目标模型。源和目标可以是相同或不同的模型。参见以下详述的“命令类别”一节。

任何模型和模型选择(具有 TContainPolicy 实施)支持来自另一模型和模型选择(具有 TContainPolicy 实施)的引用的操纵和显示。例如，一个折合库可包括对复合文件中某一特定文本部分(复合文件选择)的引用；而一个复合文件可包括对一个折合库中的特定文件(折合库选择)的引用。由于引用总是在任何模型和模型选择都知道如何吸收的 TContainableWrapperModel 中回绕，所以可以进行这种内部操作。参见下述“环绕模型类别”。

模型类别

Container Framework 提供了 TEmbedderContainerModel 它可被子分类以生成可包含和嵌入的新数据类

型的。例如,TCompoundDocumentModel 是可以支持嵌入模型以及超连接,包含和引用的文件。类似地,Container-Framework 提供了 TContainerModel,它可被子分类以生成只可以包含的新数据类型。例如,TFolderModel 是一个折合库,它可支持超连接,包含以及引用。

选择类别

Container Framework 提供了 TEmbedderContainerModelSelection,它可被进一步分类以生成可嵌入和包含的新选择类型。例如,TCompoundDocumentSelection 是一个可以支持嵌入模型以及超连接,包含和引用的文件选择。

类似地,Container Framework 提供了 TContainer-ModelSelection,它可被进一步分类以生成只能包括的新选择类型。例如,TFolderModelSelection 是可以支持超连接,包含和引用的一个折合库选择。

超连接/包含/引用类别

TModelAnchor 提供了用于管理超连接,包含和引用的下列协议:

- 属性管理协议,即加入,删除,列示和查找属性;
- 数据管理协议,即加入,复制和删除数据;
- 连接管理协议,即加入,删除,追踪,列示,和查找连接(在一个包含和一个引用的情况下只有一个连接);以及
- 可视生成协议,即生成超连接,包含和引用的可视图答。

该可视生成协议由一个复合文件表示用于生成起连接的连接图；它亦可由一个折合库表示所用以生成包含和引用的概略视图。

TContainable 是 TModelAnchor 的抽象子类别。它对包含（TOwnedContainable）和引用（TReferenced-Coutainable）的新协议进行再定义：

·属性管理协议，即检索特定属性的加速方法。这些属性可包括系统定义属性（尺寸，作者，生成日期，类型，各种图符表达以及用于生成可储存或引用的视图的装饰）。若 TContainable 在其本地高速缓冲存储器中没有发现属性，则访问模型存储器以检索属性。在某些情况下，可能在用户的界面档案中发现属性；例如，若一个模型没有定制图符，则其检索一个通用“Document”图符。数据管理协议，即对复制和删除进行重新定义的方法。

	TOwnedContainable	TReferenced Containable
复制	复制包含和模型	只复制引用
删除	删除可包含和模型	只删除引用

·在一个模型中环绕包含和引用以便用模型选择参与数据交换（即剪裁，复制，和招贴）的方法。

·用于 Preferences Fremework 的方法，用以打开在包含上的特性页。

如果需要不同的语义和 TContainable 的其他子类别或行为的话，可编写 TContainable 的其他子类别。例如，“打

开”计算机的文具(stationery)导致生成一个新文件。因此，TStationeryContainable(它是 TContainable 的子类别)必须置换其连接追随方法以提供适当的行为。

命令类别

MakeReference 命令是一个新的标准命令，它工作于任何模型选择以及其相关模型。该命令工作如下：

从源模型选择中获取数据。这些数据将具有 TModelAnchor 指针并表示一个超连接，一个包含或一个引用。

2. 在数据上生成一个引用。

3. 生成该引用的新的源模型选择。

4. 将该新的源模型选择考贝到剪取板上。

Paste 命令是现有的标准命令，它工作在任何模型选择以及其相关模型上，其工作如下：

1. 向剪裁板请求外来模型可被转换的类型的表。

2. 需求目标模型选择从该表中选择其想要外来模型被转换成的类型以及其想要接受的转换的外来模型的方式(嵌入或吸收)。

3. 向剪裁板请求转换的外来模型的复制。

4. 要求目标模型选择接受转换的外来模型。

5. 生成新模型的新目标模型选择。

环绕模型类别

需要模型利用模型选择对包含和引用进行环绕以便进行数据交换(即剪取，复制和粘贴)：该交换通货为一个模型。由于一个环绕模型既不需要 TContainPolicy 实施，亦无需

要TEmbedPolicy实施，所以其可被生成为TModel抽象类别的具体子类别。

图3是一流程图，其示出了根据本发明最佳实施例来生成一个引用目标的处理。处理始于300并立即去往功能框310，在310处，第一和第二目标被打开。目标可以是文本文件，图象，展开页表，折合库或任何其他计算机应用、程序。然后，在功能框320处，通过在显示器上选择一个角落并把一个矩形框拉伸以包含选定的数据，从而在第一目标中生成了一个选择。在功能框330处，通过示例一个锚目标而从选择中生成一个锚。该锚管理一个选择区，该选择区在被连接到锚目标期间会一直存在。功能框340说明了引用目标的生成。一个引用目标是一个环绕一个锚目标的环绕程序，它使锚目标位于任何其他目标或封闭物之内，正如引用的专利申请所讨论的那样。然后，在功能框350处，代用锚被采用在引用目标中且引用目标被用到第二目标中，如功能框360所示。代用锚是指对一个实际锚的指针。采用涉及把拥有关系一个目标传送到另一目标的必要处理。最后，处理完成于370处，虽然这一实例具有生成在一个文件并被第二个文件采用的引用目标，本技术领域的一般技术人员将识识到该引用目标可位于第一文件的不同位置处。

图4是一流程图，用于表明根据本发明的一个最佳实施例使用一个引用目标的处理。该处理始于400并立即去往功能框410，在该处打开第一目标。该目标可以是一个文本文件，图象，展开页表，折合库或其他计算机应用。然后在功能

框402处选择一个引用目标。该引用目标为一个动作抽象地指定一个目标，因为它包括一个指定该选择的锚。该动作在其后时间或查阅时间被解决。功能框430说明用于Start __side上的命令的生成处理。该处理响应用户的特定动作(例如连扣两下打开目标)。当动作发生时，将一适当命令初始化以便对该动作提供服务。因此，例如，命令可以是一个跟随或打开命令，打印机命令，删除命令，或任何与dest __side文件中的选择有关的命令。这一处理在文件的start __side生成指示的命令。然后在功能框440处，把命令送到其他文件的锚的dest __side。发送操作需要连接(如功能框450所示)，并在必要时打开dest __side文件，设置与dest __side上的锚相等同的选择，并卷动锚的位置以供浏览(如功能框460所示)。然后，用于Start __side处的命令被用于dest __side，如功能框470所示。最后，处理完成于480处。虽然这个例子具有定位于一个文件中的并在第二个文件中被引用锚的引用目标，本领域的一般技术人员将可认识到引用目标和被引用目标可位于同一文件之中。

图5示出了根据本发明的一个最佳实施例与一个远地打印机关联的显示。文件图符500当时位于文件New Request 510中。对一个远地打印机MyPrinter520的引用以图符形式启动。用于操纵一个远地源的引用目标实例示于图5,6,7以及相应的说明之中。

图6示出了根据本发明的一个最佳实施例把一个目标的考贝拖移至一个引用的示例。其中包括一个文件图符600

和该文件的轮廓 610。该轮廓是在拖移图符 600 的过程中由接下 option 键而生成的。该轮廓代表了该文件的考贝。

图 7 示出了根据本发明的一个最佳实施例部分完成一个动作中与一个引用目标相邻的目标。文件轮廓 710 被拖移到打印机引用 720 上以便在文件 700 上执行一个打印命令。

REFERENCE Program Listing
C++ Language

```
// Interface: TReferencedContainable
class TReferencedContainable : public TContainable
{
public:
        static const TContainableKind& kKind;
        // Special methods.
                                        TReferencedContainable(const
TAbstractModel&,
                                                const TAbstractModel&);
                                        TReferencedContainable();
                                        TReferencedContainable(const
TReferencedContainable&);
                                        ~TReferencedContainable();

                TReferencedContainable&         operator=(const
TReferencedContainable&);
        virtual         TStream&                operator>>=(TStream& towhere) const;
        virtual         TStream&                operator<<=(TStream& towhere);
```

```
        MCollectibleDeclarationsMacro(TReferencedContainable);
        // Presentation methods.
        virtual     TPresentableView*       CreateIconView() const;

        // Semantic methods.
        virtual     TContainableKind        GetContainableKind() const;
        virtual     void                    DeleteSelf();
        virtual     TContainable*                   DuplicateSelf() const;
};
// Interface: TMakeReferenceUserCommand
class TMakeReferenceUserCommand : public TNewAnchorCommand
{
public:
                                        TMakeReferenceUserCommand(const
TText&);
                                        TMakeReferenceUserCommand();
                                        TMakeReferenceUserCommand(const
TMakeReferenceUserCommand&);
        virtual                         ~TMakeReferenceUserCommand();

                TMakeReferenceUserCommand& operator=(const
TMakeReferenceUserCommand&);
        virtual     TStream&            operator>>=(TStream&) const;
        virtual     TStream&            operator<<=(TStream&);
        MCollectibleDeclarationsMacro(TMakeReferenceUserCommand);
public:
        virtual     Boolean             HandleLocalDo();

protected:
        virtual     Boolean             Accepts(const TModelSelection&) const;
};
//
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++++++++++
// Class Name:      TReferencedContainable
// Description:     A referenced containable is an alias to a document.
//              - CreateIconView returns an icon view for a referenced containable
presentation.
```

```
//          - DeleteSelf is overridden so that removing a referenced containable
//                  from a container only removes itself.
//          - DuplicateSelf is overriden so that copying a referenced containable
//                  in a container only copies itself.
//          - GetContainableKind returns TReferencedContainable::kKind as the type.
-------------------------------------------------------------------------------
MCollectibleDefinitionsMacro(TReferencedContainable, 0);
const TContainableKind TReferencedContainable_kKind("TReferencedContainable");
const TContainableKind& TReferencedContainable::kKind =
TReferencedContainable_kKind;
TReferencedContainable::TReferencedContainable(const TAbstractModel&
containerModel,
        const TAbstractModel& containableModel)
        : TContainable(containerModel, containableModel)
{
}
TReferencedContainable::TReferencedContainable()
        : TContainable()
{
}
TReferencedContainable::TReferencedContainable(const TReferencedContainable&
copy)
        : TContainable(copy)
{
}
TReferencedContainable::~TReferencedContainable()
{
}
TReferencedContainable& TReferencedContainable::operator=(const
TReferencedContainable& copy)
{
        if (this != &copy)
        {
                TRY
                        TContainable::operator=(copy);
                CATCH(TStandardException)
                        RETHROW *this;
```

```
            ENDTRY
        }
        return (*this);
}
TStream&
TReferencedContainable::operator>>=(TStream& towhere) const
{
        TRY
            MAbstractModelAnchor::WriteVersion(towhere);
            TContainable::operator>>=(towhere);
        CATCH(TStandardException)
            RETHROW towhere;
        ENDTRY
        return towhere;
}
TStream&
TReferencedContainable::operator<<=(TStream& fromwhere)
{
        TRY
            VersionInfo theVersion =
MAbstractModelAnchor::ReadVersion(fromwhere);
            TContainable::operator<<=(fromwhere);
        CATCH(TStandardException)
            RETHROW fromwhere;
        ENDTRY
        return fromwhere;
}
TPresentableView*
TReferencedContainable::CreateIconView() const
{
        // create the icon view for the referenced containable.
        TPresentableView* theView = new TContainableIconView((TContainable*)this,
/*isReference=*/ TRUE);
            return (theView);
}
TContainableKind
TReferencedContainable::GetContainableKind() const
{
```

```
	return (TReferencedContainable::kKind);
}
void
TReferencedContainable::DeleteSelf()
{
}
TContainable*
TReferencedContainable::DuplicateSelf() const
{
	return ((TContainable*)this);
}
//
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
// Class Name:	TMakeReferenceUserCommand
// Description: TMakeReferenceUserCommand is activated by a user selecting a menu
item.
//		It constructs a reference to the document represented by a containable
//		and adds the reference to the container model. Currently,
//		this command is a subclass of TNewAnchorCommand and depends on
the
//		current implementation of a containable (referenced or owned) being
//		a model anchor. It reuses HandleReDo, HandleUndo,
HandleNewAnchor
//		of TNewAnchorCommand to add the reference to the container model.
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
MCollectibleDefinitionsMacro(TMakeReferenceUserCommand, 0);
const TStandardText kMakeReferenceMenuName("Make Reference");
TMakeReferenceUserCommand::TMakeReferenceUserCommand(const TText&
theCommandName)
	: TNewAnchorCommand()
{
	SetName(theCommandName);
}
TMakeReferenceUserCommand::TMakeReferenceUserCommand()
	: TNewAnchorCommand()
{
```

```
        SetName(kMakeReferenceMenuName);
}
TMakeReferenceUserCommand::TMakeReferenceUserCommand(const
TMakeReferenceUserCommand& theCommand)
        : TNewAnchorCommand(theCommand)
{
        SetName(kMakeReferenceMenuName);
}
TMakeReferenceUserCommand::~TMakeReferenceUserCommand()
{
}

TMakeReferenceUserCommand&
TMakeReferenceUserCommand::operator=(const TMakeReferenceUserCommand&
theCommand)
{
        if (this != &theCommand)
        {
                TRY
                        TNewAnchorCommand::operator=(theCommand);
                CATCH(TStandardException)
                        RETHROW *this;
                ENDTRY
        }
        return *this;
}
TStream&
TMakeReferenceUserCommand::operator>>=(TStream& towhere) const
{
        TRY
                WriteVersion(towhere);
                TNewAnchorCommand::operator>>=(towhere);
        CATCH(TStandardException)
                RETHROW towhere;
        ENDTRY
        return towhere;
}
TStream&
```

```
TMakeReferenceUserCommand::operator<<=(TStream& fromwhere)
{
	TRY
		VersionInfo theVersion = ReadVersion(fromwhere);
		TNewAnchorCommand::operator<<=(fromwhere);
	CATCH(TStandardException)
		RETHROW fromwhere;
	ENDTRY
	return fromwhere;
}
Boolean
TMakeReferenceUserCommand::Accepts(const TModelSelection& theSelection) const
{
	Boolean accept = TModelCommand::Accepts(theSelection);

	if (accept)
	{
		accept = !theSelection.IsEmpty(); // command is enabled if selection is
non-empty
	}

	return (accept);
}
Boolean
TMakeReferenceUserCommand::HandleLocalDo()
{
	const TWorkspaceContainerSelection* theSelection = (const
TWorkspaceContainerSelection*)GetBoundSelection();
	TContainableSurrogateIterator anIterator(theSelection);
	TModelAnchorSurrogate* theSurrogate = anIterator.First();
	if (theSurrogate != NIL)
	{
		TDocumentEntry entry;
		TModel* theModel = theSelection->GetModel();
		TContainable* theContainable = (TContainable*)theModel-
>LookupAnchor(*theSurrogate);
		const TModelLink* theLink = theContainable->GetLink();
```

```
            TModelSurrogate* containerSurrogate = theLink->GetHere()-
>CopyModelSurrogate();
            TModelSurrogate* containableSurrogate = theLink->GetThere()-
>CopyModelSurrogate();
            // construct a reference whose containerSurrogate and
containableSurrogate
            // equals those of the containable it refers to.
            TContainable* aReference = new
TReferencedContainable(*containerSurrogate, *containableSurrogate);
            // construct a default selection with the reference as a member.
            TModelSelection* referenceSelection = theModel-
>CreateDefaultSelection();
            referenceSelection->AddAnchor(*aReference);
            // have the reference adopt the selection.
            aReference->AdoptSelection(referenceSelection);

            // if the referenced containable is a stationery, then make the reference a
stationery
            if (theContainable->IsStationery())
                  aReference->ToggleStationery();

            // have the container model adopt the reference.
            AdoptAnchor(aReference);
      }
      return (TRUE);
}
```

虽然本发明已根据一特定的系统环境对最佳实施例进行了叙述，但本技术领域的技术人员可认识到可在本发明权利要求限定的范围和本发明的概念范围内对本发明进行修改或在其他不同的软硬件环境中实施本发明。因此，本发明的保护范围由权利要求书所涵盖。

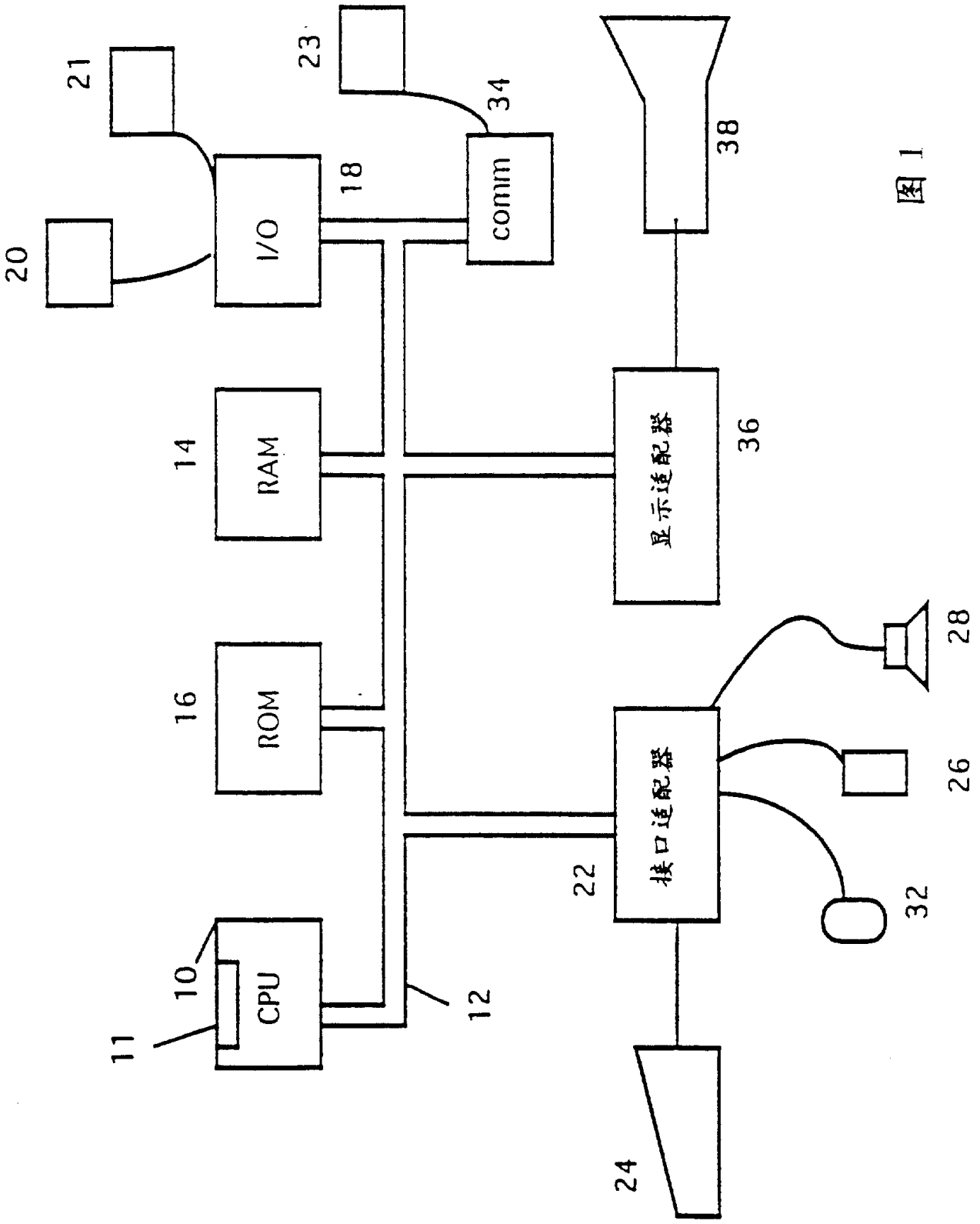

CPU
10
11
12
ROM
16
RAM
14
I/O
18
20
21
comm
34
23
显示适配器
36
38
接口适配器
22
24
26
28
32

图 1

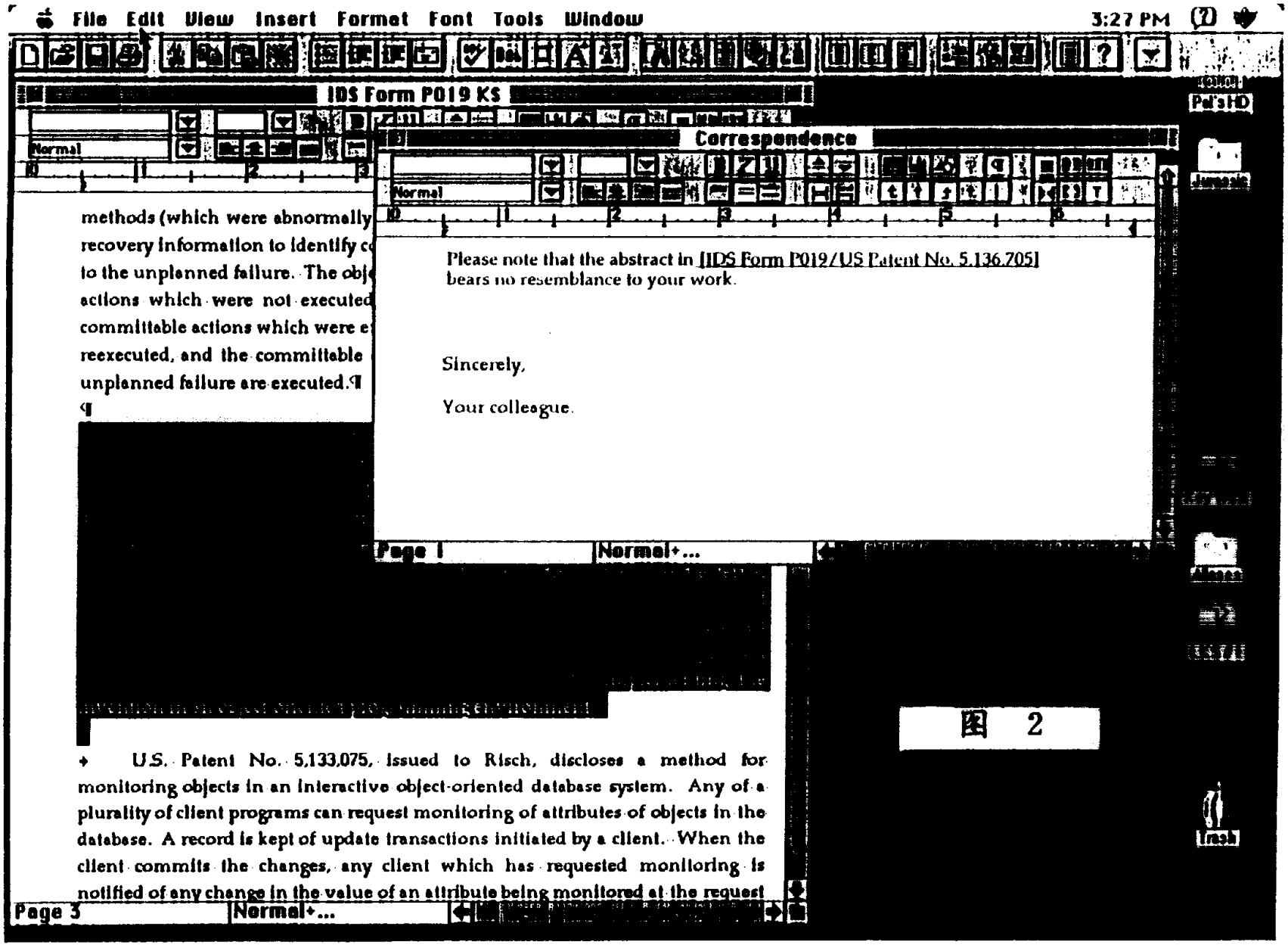

File Edit View Insert Format Font Tools Window
3:27 PM
IDS Form P019 KS
Correspondence
Normal
Please note that the abstract in [IDS Form P019/US Patent No. 5,136,705] bears no resemblance to your work.
Sincerely,
Your colleague.
Page 1
Normal+...
methods (which were abnormally
recovery information to identify c
to the unplanned failure. The obj
actions which were not executed
committable actions which were e
reexecuted, and the committable
unplanned failure are executed.
U.S. Patent No. 5,133,075, issued to Risch, discloses a method for monitoring objects in an interactive object-oriented database system. Any of a plurality of client programs can request monitoring of attributes of objects in the database. A record is kept of update transactions initiated by a client. When the client commits the changes, any client which has requested monitoring is notified of any change in the value of an attribute being monitored at the request
Page 3
Normal+...
Trash

図 2

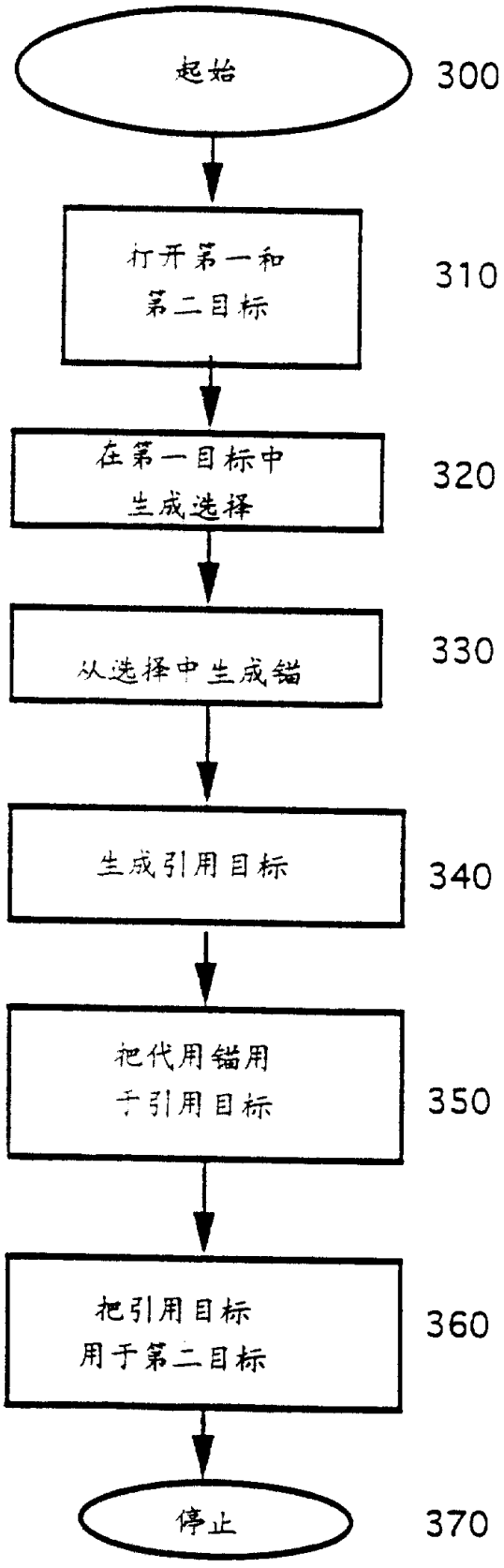

起始
300
打开第一和
第二目标
310
在第一目标中
生成选择
320
从选择中生成锚
330
生成引用目标
340
把代用锚用
于引用目标
350
把引用目标
用于第二目标
360
停止
370

图 3

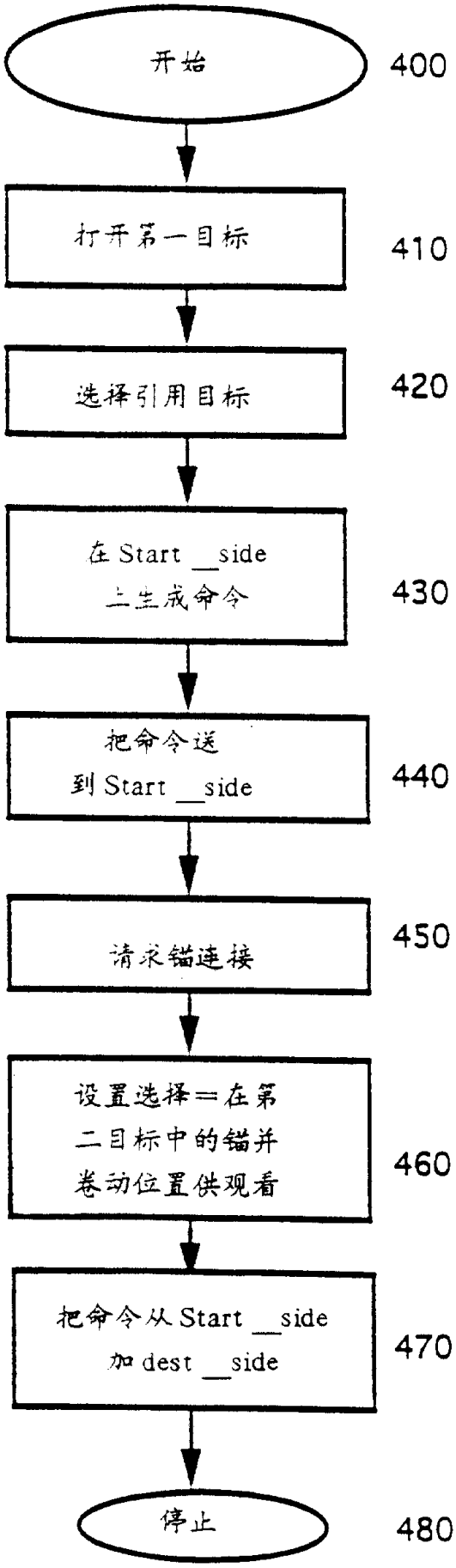

开始
400
打开第一目标
410
选择引用目标
420
在 Start __side
上生成命令
430
把命令送
到 Start __side
440
请求锚连接
450
设置选择＝在第
二目标中的锚并
卷动位置供观看
460
把命令从 Start __side
加 dest __side
470
停止
480

图 4

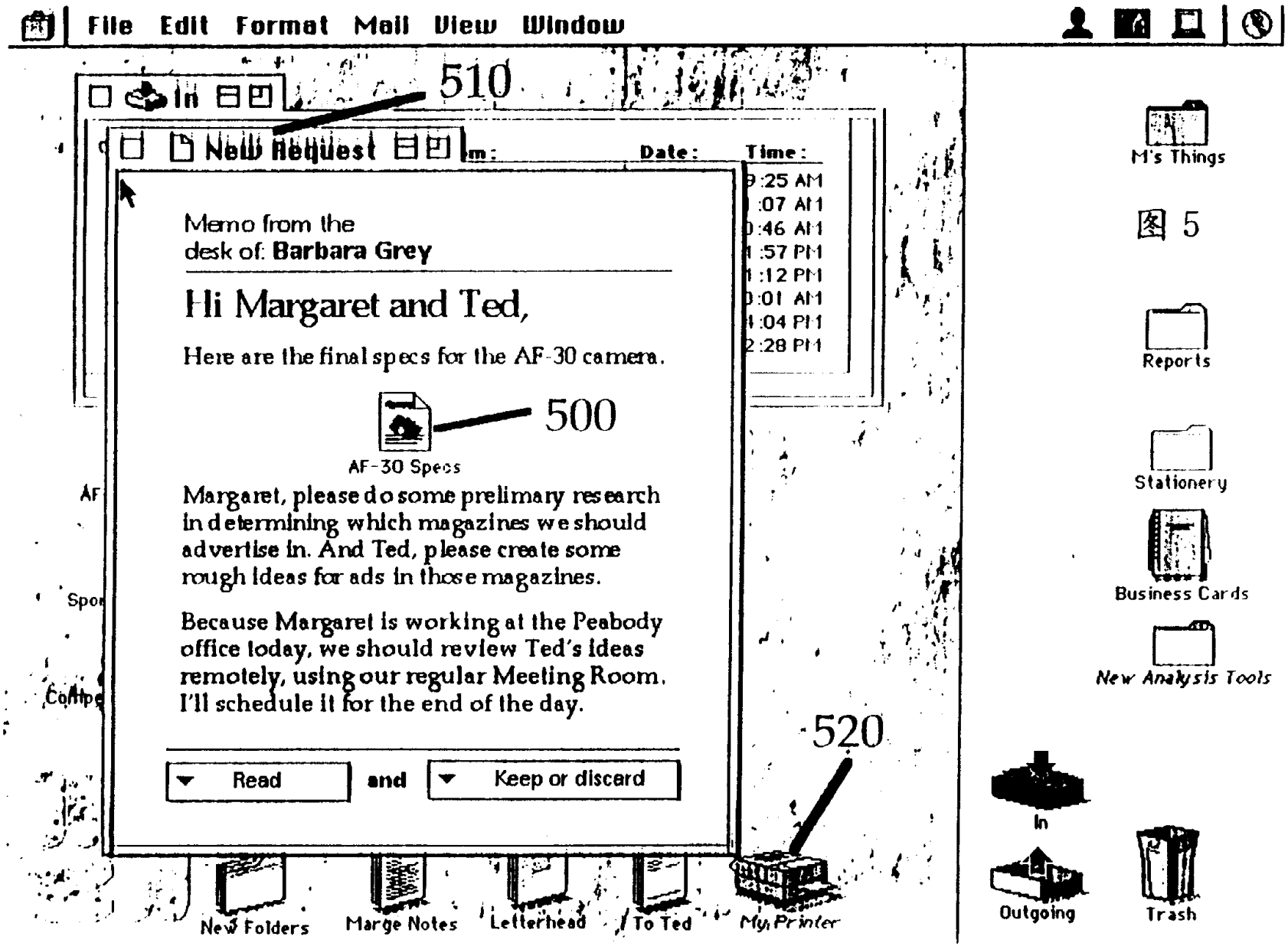

File Edit Format Mail View Window
In
510
New Request
m:
Date:
Time:
:25 AM
:07 AM
:46 AM
:57 PM
:12 PM
:01 AM
:04 PM
:28 PM
Memo from the
desk of: Barbara Grey
Hi Margaret and Ted,
Here are the final specs for the AF-30 camera.
500
AF-30 Specs
Margaret, please do some prelimary research in determining which magazines we should advertise in. And Ted, please create some rough ideas for ads in those magazines.
Because Margaret is working at the Peabody office today, we should review Ted's ideas remotely, using our regular Meeting Room. I'll schedule it for the end of the day.
Read
and
Keep or discard
520
New Folders
Marge Notes
Letterhead
To Ted
My Printer
M's Things
Reports
Stationery
Business Cards
New Analysis Tools
In
Outgoing
Trash

图 5

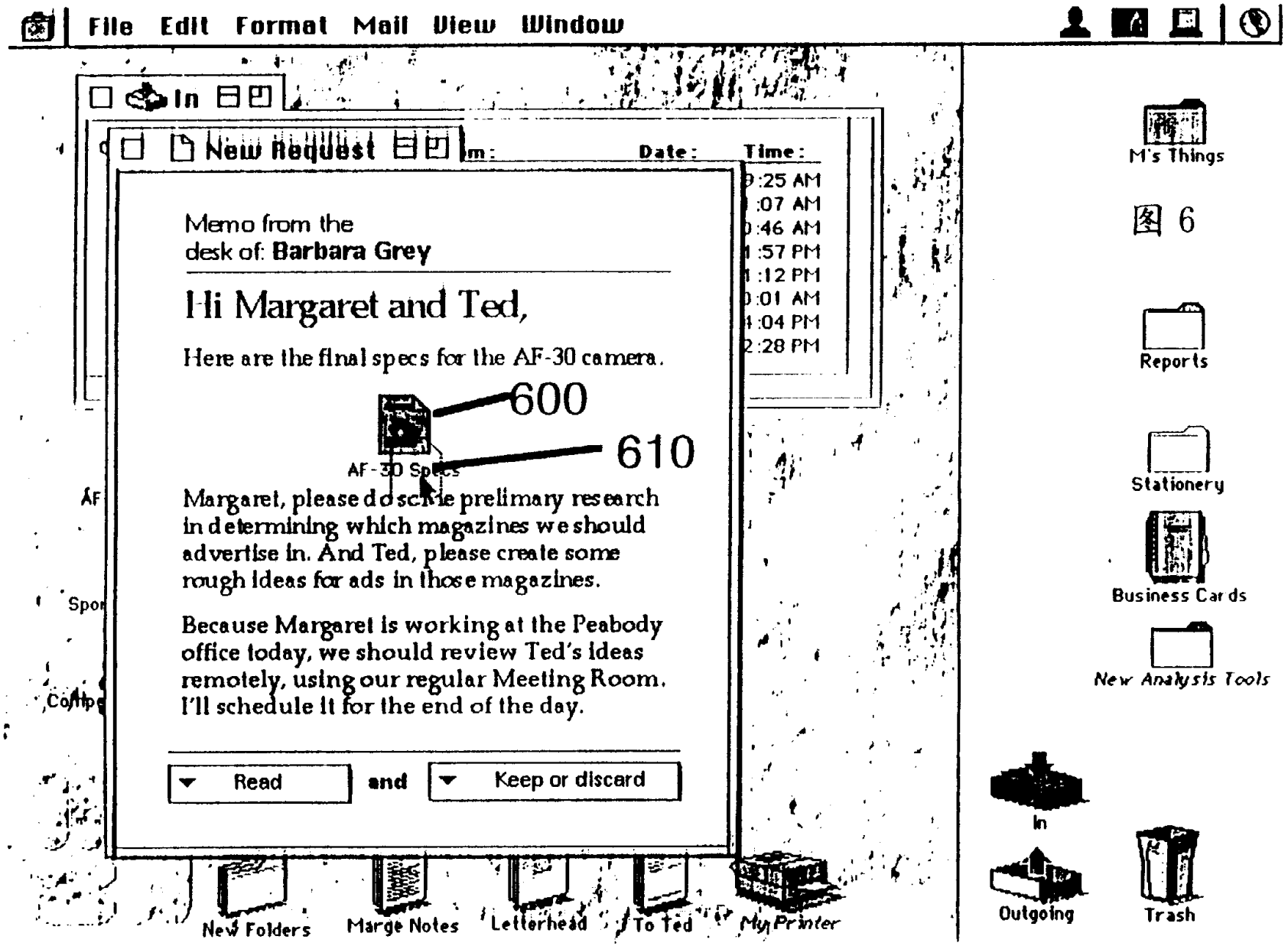

File Edit Format Mail View Window
In
New Request
m:
Date:
Time:
:25 AM
:07 AM
:46 AM
:57 PM
:12 PM
:01 AM
:04 PM
:28 PM
Memo from the
desk of: Barbara Grey
Hi Margaret and Ted,
Here are the final specs for the AF-30 camera.
600
610
AF-30 Specs
Margaret, please do some prelimary research in determining which magazines we should advertise in. And Ted, please create some rough ideas for ads in those magazines.
Because Margaret is working at the Peabody office today, we should review Ted's ideas remotely, using our regular Meeting Room. I'll schedule it for the end of the day.
Read
and
Keep or discard
New Folders
Marge Notes
Letterhead
To Ted
My Printer
M's Things
Reports
Stationery
Business Cards
New Analysis Tools
In
Outgoing
Trash

图 6

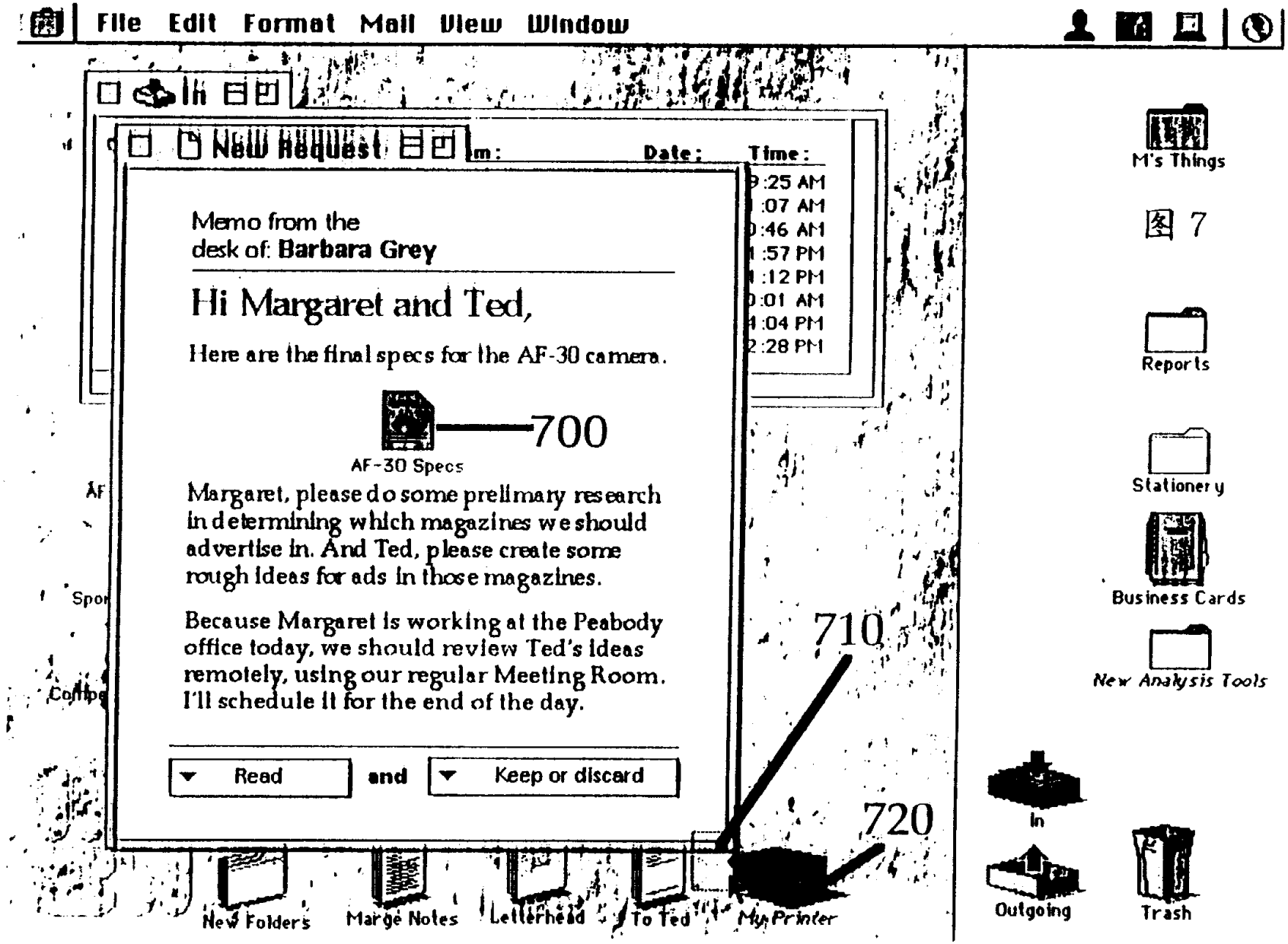

File Edit Format Mail View Window
New Request
Date:
Time:
:25 AM
:07 AM
:46 AM
:57 PM
:12 PM
:01 AM
:04 PM
:28 PM
Memo from the
desk of: Barbara Grey
Hi Margaret and Ted,
Here are the final specs for the AF-30 camera.
700
AF-30 Specs
Margaret, please do some prelimary research in determining which magazines we should advertise in. And Ted, please create some rough ideas for ads in those magazines.
Because Margaret is working at the Peabody office today, we should review Ted's ideas remotely, using our regular Meeting Room. I'll schedule it for the end of the day.
Read
and
Keep or discard
710
720
M's Things
Reports
Stationery
Business Cards
New Analysis Tools
In
Outgoing
Trash
New Folders
Marge Notes
Letterhead
To Ted
My Printer

图 7