

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.  
G06F 12/06 (2006.01)



# [12] 发明专利说明书

专利号 ZL 02102374.3

[45] 授权公告日 2007 年 7 月 11 日

[11] 授权公告号 CN 1326049C

[22] 申请日 2002.1.18 [21] 申请号 02102374.3  
 [30] 优先权  
     [32] 2001. 1. 18 [33] EP [31] 01300430.4  
 [73] 专利权人 索尼服务中心(欧洲)股份有限公司  
     地址 比利时隆德泽尔  
 [72] 发明人 L·斯卡林格 O·蒂里  
 [56] 参考文献  
     US5619250A 1997.4.8  
     CN1277778A 2000.12.20  
 审查员 林松岭

[74] 专利代理机构 中国专利代理(香港)有限公司  
 代理人 吴增勇 傅康

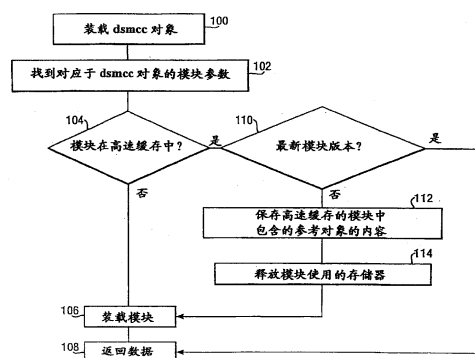
权利要求书 2 页 说明书 10 页 附图 4 页

## [54] 发明名称

将下载的对象提供给应用程序的方法和装置

## [57] 摘要

一种用于将下载的对象提供给应用程序的方法和装置，包括从接收传输流下载包含应用程序请求的对象的模块；存储下载的模块；监视存储的模块是否为最近的版本；当应用程序请求存储的模块的对象，而存储的模块是最近的版本时，将存储的模块的对象提供给应用程序；以及当应用程序请求存储的模块的对象，而存储的模块不是最近的版本时，仅复制应用程序仍在使用的被存储模块的对象，下载最近版本的被存储模块，并用下载的最近版本的被存储模块来重写所述被存储模块。



1. 一种用于把下载对象提供给应用程序的装置，所述装置包括：  
对象层接口，用于从接收传输流下载包含应用程序请求的对象的模块；  
模块存储器，用于存储下载的模块；以及  
控制器，用于监视所述模块存储器中的所述模块是否为最近的版本；  
其中

当所述应用程序请求存储的模块中包含的对象，而所述存储的模块是最近的版本时，所述对象层接口将所述模块存储器的所述对象提供给所述应用程序，并且当所述应用程序请求存储的模块中包含的对象，而所述存储的模块不是最近的版本时，所述对象层接口下载所述最近的版本的所述模块，并重写所述模块存储器中存储的所述模块；所述装置还包括：

对象高速缓存，用于存储对象；其中

当所述应用程序请求存储的模块中包含的对象，而所述存储的模块不是所述最近的版本时，在所述对象层接口用所述最近的版本的所述存储的模块来重写所述存储的模块之前，所述控制器仅将所述应用程序仍在使用的所述存储的模块的所述对象复制到所述对象高速缓存。

2. 如权利要求1所述的装置，其特征在于所述控制器对所述传输流中有关所述传输流中传送的所述最近版本的所述模块的信息进行响应。

3. 如权利要求1或2所述的装置，其特征在于所述控制器按照多媒体家庭平台规范来监视模块是否为所述最近版本。

4. 如权利要求1或2所述的装置，其特征在于所述对象是DSM-CC对象，并且所述模块是DSM-CC模块。

5. 如权利要求4所述的装置，其特征在于通过复制对应的BIOP消息来把所述对象从所述模块存储器复制到所述对象高速缓存。

6. 如权利要求1或2所述的装置，其特征在于还包括至少一个过滤器，用于从所述传输流滤出要下载的模块的信息包。

7. 如权利要求1或2所述的装置，其特征在于所述装置是电视装置。

8. 如权利要求1或2所述的装置，其特征在于所述装置是多媒体家庭

平台规范顺应的。

9. 一种将下载的对象提供给应用程序的方法，所述方法包括：

从接收传输流下载包含应用程序请求的对象的模块；

存储所述下载的模块；

监视所述存储的模块是否为最近的版本；

当所述应用程序请求存储的模块的对象，而所述存储的模块是最近的版本时，将所述存储的模块的所述对象提供给所述应用程序；以及

当所述应用程序请求存储的模块的对象，而所述存储的模块不是最近的版本时，仅复制所述应用程序仍在使用的所述存储的模块的所述对象，下载所述最近版本的所述存储的模块，并用所述下载的最近版本的所述存储的模块来重写所述存储的模块。

## 将下载的对象提供给应用程序的方法和装置

### 技术领域

本发明涉及一种用于将下载的对象提供给应用程序的方法和装置，具体地说，涉及从传输流下载信息以便按请求把数据对象提供给应用程序。

### 背景技术

众所周知，在传输流中传送数据对象供终端用户下载。对象可包括供终端用户使用的任何形式的数据，并且可以分布式重复形式传送，例如，通过数据轮放（data carousel）传送。数据对象常可以一起被分组为数据模块，这样，对于要使用特定对象的终端用户的应用程序，要从传输流下载包含该对象的整个模块。这些情况下，服务提供商可以将相关对象一起分组在各个模块中，这样，如果终端用户存储下载模块，则大大减少了对相关对象的访问时间。

按照 DSM-CC 广播协议，诸如文件、目录等的对象以 DSM-CC 模块内承载的 BIOP 消息的形式传送到终端用户的接收机。这些模块可通过轮放被广播，每个模块重新格式化为多个 MPEG 段（MPEG section），这些段以分布形式被广播供终端用户接收并将其重新组合成各个模块。

随着时间的推移，服务提供商将更新对象。在这些情况下，最好是，当终端用户的应用程序请求对象时，它将收到该对象的最新版本。然而，如果在更新对象前，终端用户已经下载了相应的模块，则将终端用户存储的模块中包含的旧版本对象提供给应用程序。

### 发明内容

本发明的一个目的是克服上述的问题。

按照本发明，提供了一种用于把下载对象提供给应用程序的装置，所述装置包括：对象层接口，用于从接收传输流下载包含应用程序请求的对象的模块；模块存储器，用于存储下载模块；以及控制器，用于监视所述模块存储器中的所述模块是否为最近的版本；其中当所述应用程序请求存储的模块中包含的对象，而所述存储的模块是最近的版本时，所述对象

层接口将所述模块存储器的所述对象提供给所述应用程序，并且当所述应用程序请求存储的模块中包含的对象，而所述存储的模块不是最近的版本时，所述对象层接口下载所述最近的版本的所述模块，并重写所述模块存储器中存储的所述模块；所述装置还包括：对象高速缓存，用于存储对象；其中当所述应用程序请求存储的模块中包含的对象，而所述存储的模块不是所述最近的版本时，在所述对象层接口用所述最近的版本的所述存储的模块来重写所述存储的模块之前，所述控制器仅将所述应用程序仍在使用的所述存储的模块的所述对象复制到所述对象高速缓存。

本发明还提供了一种将下载的对象提供给应用程序的方法，所述方法包括：从接收传输流下载包含应用程序请求的对象的模块；存储所述下载的模块；监视所述存储的模块是否为最近的版本；当所述应用程序请求存储的模块的对象，而所述存储的模块是最近的版本时，将所述存储的模块的所述对象提供给所述应用程序；以及当所述应用程序请求存储的模块的对象，而所述存储的模块不是最近的版本时，仅复制所述应用程序仍在使用的所述存储的模块的所述对象，下载所述最近版本的所述存储的模块，并用所述下载的最近版本的所述存储的模块来重写所述存储的模块。

这样，在应用程序请求对象时，始终提供有该对象的最新版本。然而，应用程序仍可以继续使用以前请求的相同模块中的任何对象。

另一方面，在应用程序请求存储的模块中包含的对象，而该存储的模块不是最近的版本，但包含了所述应用程序已在使用的其它对象时，所述装置可只下载并存储所述模块的新的最近版本，并且是除以前版本的模块之外又存储下载的最近版本的模块。这样，应用程序将访问最近版本的新请求对象，同时保持与以前请求的对象的交互作用。然而，此方法非常浪费存储器，特别是在模块包含了许多对象而应用程序可能只使用存储的模块的一个对象的情况下。

同样地，装置有可能只包括替代模块存储器的对象存储器，因而它只存储应用程序在使用的对象。这样，每当应用程序请求新对象时，终端用户始终从传输流下载对象，因此始终获得最近的版本。然而，在相关对象一起包含在各个模块中时，常常会从以前存储的模块检索请求对象，此方法严重限制了访问速度。

因此，本发明允许以最佳的访问时间但却以最低的存储器要求将对象提供给应用程序。

由于应用程序的请求是以这样的方式被处理，因此应用程序本身不需要知道它使用的对象源。通过硬件或软件实现的对象层可实现对象的透明高速缓存。

最好是，控制器对传输流中有关传输流中传送的最近版本的模块的信息进行响应。

在诸如从 DSM-CC 轮放广播的许多系统中，在传输流中传送有关轮放上模块更新的信息。控制器可以监视此信息，这样，在应用程序请求更新模块的对象时，控制器可采取上述定义的步骤。

最好是，控制器按照 MHP 平台的规范来监视模块是否为最近的版本。

因此，在按照 MHP 规范的广播环境中，可提供对象的透明高速缓存，这样，始终将最新的对象提供给应用程序。

最好是，对象为 DSM-CC 对象且模块为 DSM-CC 模块。

因此，本发明可与 DSM-CC 系统结合使用。

最好是，通过复制对应的 BIOP 消息，将对象从模块存储器复制到对象高速缓存。由于 DSM-CC 对象作为 BIOP 消息包含在 DSM-CC 模块中，因此对象高速缓存可以按其 BIOP 消息形式存储对象。

最好是，提供至少一个过滤器，用于滤出要下载的模块的传输流信息包。这可以通过滤出要下载的模块的段来实现，例如，MPEG-2 段。因此，检查传输流中诸如 MPEG 信息包的信息包来确定它们是否涉及要下载的模块。由段过滤器从传输流复制所有相关的信息包，这样，可以将它们重新装配以形成模块。

最好是，装置为电视装置。因此，本发明可实施在接收其它电视服务的电视装置中，诸如实施在集成电视和机顶盒中。

最好是，装置是 MHP 顺应的。

可以理解，本发明可用软件来实现并记录在存储媒体上。实际上，已经配置成对请求对象的模块进行下载的装置可由适当的对象层进行补充，这样，按照本发明，当应用程序请求存储的模块的对象，而该存储的模块不是最近的版本时，复制所述应用程序仍在使用的所述存储的模块的对

象。对象层提供透明高速缓存，这样，应用程序可以继续使用复制的对象而无需了解复制或存储器寻址中的变化。

### 附图说明

参照附图，通过仅以示例方式给出的以下描述，可更好地理解本发明，附图中：

图 1 说明基于模块的 DSM-CC 客户栈；

图 2 示意性说明实施本发明的装置；

图 3 说明实施本发明的过程的流程图；以及

图 4 (a)、(b)、(c) 和 (d) 说明更新情况。

### 具体实施方式

现将参考用于按照 DSM-CC 广播协议的实施例来描述本发明。然而，可以理解，只要系统中用于终端用户应用程序的对象在模块中一起被传送，本发明便适用于这样的系统。

按照 DSM-CC 广播协议，诸如文件和目录的对象以 BIOP 消息的形式被传送到终端用户接收机，它们本身被承载在数据轮放模块中。一个这样的模块可以并且通常包含几个 BIOP 消息。在这方面，服务提供商通常将相关 BIOP 消息一起分组在相同的模块中。

因此，由于 DSM-CC 对象轮放的这种特性，DSM-CC 客户栈，即在接收机中实现的处理下载数据的软件层栈，常常实现为“基于模块的”。换而言之，把从接收传输流下载的数据保持在模块层，即以接收模块的形式而不是作为原对象或 BIOP 消息来保持。

通过将接收数据以接收模块的形式存储在模块层，可以提高终端用户装置得到的访问速度。特别是，由于相关对象一般一起被分组在相同模块中，因而最好将整个模块保持在存储器中。这是因为已请求模块的一个对象的用户应用程序可能请求相同模块中的其它对象。如果该模块保持在存储器中，则与要从传输流进行检索相比，应用程序显然能以更快的速度访问它。

图 1 说明用于如上所述操作的基于模块的 DSM-CC 客户栈。

可以运行终端用户应用程序的客户层 2 能够通过诸如索引号码、数据地址等的不透明参考来请求 DSM-CC 对象。

随后，用户到用户层 4 提供所有 BIOP 消息的高级表示。此层执行客户应用程序可调用的所有操作，诸如读取文件，列出目录等。它与对象轮放层连接，以便按照客户层中应用程序对对象的请求来请求装载（或卸载）BIOP 消息。

对象轮放层 6 维护所有 BIOP 消息并向用户到用户层 4 提供基本 BIOP 消息访问例程序。为了按照客户层 2 中应用程序的需要来获得适当的 BIOP 消息，对象轮放层 6 与数据轮放层 8 连接，以便请求装载或关闭数据轮放模块。

数据轮放层 8 是客户栈的最低层，并负责装配数据轮放模块。因此，在需要特定模块以将请求的对象提供给客户层 2 中的应用程序时，数据轮放 8 设置适当的段过滤器，以对承载与所需模块有关的 DSM-CC 消息的 MPEG 2 段进行抽取。在不再需要过滤器时，数据轮放层 8 也可消除过滤器。

为将某特定对象提供给应用程序而从传输流下载模块时存储整个该模块的方法可描述为“基于模块的高速缓存”。在处理动态内容，即处理内容随时间而变的对象时，此方法产生问题。特别是，在应用程序请求一个对象，而该对象包含在已经可以在装置的高速缓存中得到的模块中时，如果该对象的内容易受变化，则无法保证高速缓存中的对象为最近的版本。

显然希望应用程序始终接收最新版本的对象。实际上，按照 MHP 规范，DSM-CC 栈必须返回对应于传输流中最新可用版本的对象。这种行为称为“透明高速缓存”。为实现透明高速缓存，已经存储在高速缓存中的模块似乎在任一给定时间点需要从广播传输流重新装载。在已经存在参考该模块的情况下，问题产生，例如客户层 2 的应用程序正在使用该模块的对象中的至少一个，这些参考必须仍保持有效。因此，如果客户层 2 的应用程序正在使用模块的一个或多个对象，则应用程序必须能够继续使用那些没有变化的对象的数据。特别是，如果应用程序已根据特定对象中包含的数据正在运行，则不允许在应用程序执行操作的中途改变该数据。因此，当从广播流装载新版本时似乎需要保持旧版本的模块。

通过使数据轮放层 8 保持几个版本的相同模块，出现了存储器效率的大量降低问题，因为不在使用的过时对象数据必须毫无必要地保持在存储



器中。此外，采用这种配置，客户栈较低层的复杂性（数据轮放客户实现）将需要增加。

另外，DSM-CC 客户能以基于对象的方式实现。换言之，只是高速缓存由客户层 2 中应用程序请求的对象，而不是高速缓存装载的模块。这被描述为基于对象的高速缓存方法。这将易于迎合上述透明高速缓存。然而，由于对象的所有新参考将需要从广播流来下载模块，因此客户层 2 的应用程序的操作将会慢得多。

图 2 以示意性示出可克服这些问题的装置。当然，实际上此装置可能用软件实现，具有参照图 1 所述的客户栈的形式。

如下所述的那样，DSM-CC 客户实施为上述两种高速缓存方法之间的混合。特别是，只要发现模块是旧版本，换言之是模块已过时，DSM-CC 栈便仅复制需要保持的数据，即被参考的对象或者实际上更可能是 BIOP 消息，然后释放该模块。

按照应用程序 10 对特定对象的请求，对象层接口 12 搜索模块存储器 14，查找包含请求对象的 BIOP 消息的对应模块。实际上，可保持被存储的以前下载的模块的记录。

如果模块存储器 14 中尚未存储适当的模块，则采取措施从接收广播传输流 15 来下载所需数据。特别是，可使用一个或多个段过滤器 16 从流 15 抽取重构所要模块所需的段。例如，可使用 MPEG 段过滤器抽取承载所需 DSM-CC 模块的 MPEG-2 段。

当所要的模块被接收时，由对象层接口 12 将该模块中包含的原请求对象提供给应用程序 10。整个模块也存储在模块存储器 14 中。

还提供控制器 18。可以理解，这是可包含在装置的整个操作中的功能的示意性表示。

控制器 18 对广播传输流 15 中的数据反应敏感，并确定何时更新了通过流 15 可获得的模块。这样，控制器 18 可确定存储在模块存储器 14 中的模块是最近的版本还是旧版本。如所示的那样，控制器 18 直接从广播传输流 15 得到此版本信息。然而，如上所述，此表示是示意性的，实际上，此信息可通过适当的段过滤器来得到。

下面将更详细的讨论，当控制器确定存储在模块存储器 14 中的模块

现在可获得新版本时，它可以开始从广播传输流 15 下载该新版本。一检测到新版本或就可以开始所述下载，或者一旦对象层接口 12 尝试访问该模块中包含的对象便可以开始所述下载。

应用程序已在使用要更新的模块中包含的一个或多个对象时，控制器 18 先将对象或对应的 BIOP 消息复制到对象高速缓存 20。对象层接口 12 被告知此复制，这样，应用程序可以继续使用所述对象的数据而无需了解数据传送。换言之，对象层接口可以处理所需的任何重新寻址，这样，应用程序 10 是从对象高速缓存 20 而不是从模块存储器 14 来检索数据。

这样，当前在使用的所有对象均保持在对象高速缓存中，而模块被更新为新的最近版本，因此，对包含在这些模块中的任何对象的新参考始终是对最新版本的对象的参考。

如上所述，一出现新版本的模块，控制器就可以更新模块存储器，或者可以只是在应用程序 10 尝试参考该模块中的对象时更新该模块。应用程序 10 已在参考模块中至少一个对象，并因此可能在以后要参考该模块的其它对象的情况下，只要可获得新的版本，便可更新模块。通过一可获得新版本便更新模块，可能会减少将来的访问时间。另一方面，在一些情况下，每当出现新版本便更新模块可视为毫无必要的处理，这是因为应用程序 10 可能不需要参考该模块的任何其它对象。

因此，如果在更新模块，则仍被打开的对象将在对象层而不是在模块层被高速缓存。在给定时间，只有模块的最新可获得版本将保持于系统存储器中。这意味着只有重要的，即可能有用的数据由 DSM-CC 栈始终保持在高速缓存中。这样，该机制组合了基于模块的高速缓存的速度和基于对象的高速缓存的存储器效率。客户层 2 和客户层 2 中的任一应用程序 10 从不直接访问对象数据，因为对象或 BIOP 消息复制到对象高速缓存 20 将改变存储器中的对象数据位置。对象层接口 12 表示的对象轮放层 6 提供保证此行为的隔离层。

该过程在图 3 的流程图中说明。

在步骤 100 中，产生装载 DSM-CC 对象的请求。结果，在步骤 102 中为对应于请求 DSM-CC 对象的模块找到模块参数。这样，系统可以保持合适的参数数据库，以便确定所需模块。

在步骤 104 中，确定模块是否在高速缓存中，例如它是否存储在模块存储器 14 中。

如果以前尚未下载且因此未存储模块，则过程进行到步骤 106，在该步骤中，将模块装载并存储在高速缓存中。随后，在步骤 108 中，请求 DSM-CC 对象的所需数据被返回到应用程序。

如果在步骤 104 中，确定所需的模块不在高速缓存中，则过程进行到步骤 110。在此步骤中，确定高速缓存中的所要模块是否为该模块的最新版本。如果是，则过程可进行到步骤 108 并将对象数据返回给应用程序。

如果步骤 110 确定高速缓存中的模块不是最新版本，则过程进行到步骤 112。在此步骤中，高速缓存的模块中包含的所有参考对象，即应用程序已参考的对象，随后被保存，例如，将它们复制到对象高速缓存 20。

然后，在步骤 114 中，模块使用的存储器被释放，这样，过程会进行到步骤 106 以装载所需的模块。

应该知道，如上所述，过程可以另外连续执行步骤 110、112、114 和 106，以确保始终可获得下载的模块的最新版本，而不是等待对模块对象的请求。

现在将参照图 4 (a) 到 (d) 来描述特定情况。

在此情况中，应用程序有对两个对象的打开参考，这两个对象在相同模块中传送。两个对象的版本均为 N。当前可在流上获得的版本为 N+1。然而，此时，应用程序不知道这点。

图 4 (a) 中，说明应用程序通过版本为 N 的所需模块的 BIOP 消息 1 和 BIOP 消息 2 来参考对象 1 和对象 2。

图 4 (b) 中，应用程序请求对象 3。此时，数据轮放层检查可在流上获得的版本，并发现版本 N+1。

如图 4 (c) 所示，数据轮放层通知对象轮放层可在流上获得新版本的模块。对象轮放层因此从模块复制其参考的所有 BIOP 消息并关闭所有对它的访问。

现在可安全地装载新版本的模块，并且如图 4 (d) 所示，在装载完成时，对象 3 现在访问最新的版本，而对象 1 和对象 2 仍访问以前的版本。

或 BIOP 消息复制到对象高速缓存 20 将改变存储器中的对象数据位置。对象层接口 12 表示的对象轮放层 6 提供保证此行为的隔离层。

该过程在图 3 的流程图中说明。

在步骤 100 中，产生装载 DSM-CC 对象的请求。结果，在步骤 102 中为对应于请求 DSM-CC 对象的模块找到模块参数。这样，系统可以保持合适的参数数据库，以便确定所需模块。

在步骤 104 中，确定模块是否在高速缓存中，例如它是否存储在模块存储器 14 中。

如果以前尚未下载且因此未存储模块，则过程进行到步骤 106，在该步骤中，将模块装载并存储在高速缓存中。随后，在步骤 108 中，请求 DSM-CC 对象的所需数据被返回到应用程序。

如果在步骤 104 中，确定所需的模块不在高速缓存中，则过程进行到步骤 110。在此步骤中，确定高速缓存中的所要模块是否为该模块的最新版本。如果是，则过程可进行到步骤 108 并将对象数据返回给应用程序。

如果步骤 110 确定高速缓存中的模块不是最新版本，则过程进行到步骤 112。在此步骤中，高速缓存的模块中包含的所有参考对象，即应用程序已参考的对象，随后被保存，例如，将它们复制到对象高速缓存 20。

然后，在步骤 114 中，模块使用的存储器被释放，这样，过程会进行到步骤 106 以装载所需的模块。

应该知道，如上所述，过程可以另外连续执行步骤 110、112、114 和 106，以确保始终可获得下载的模块的最新版本，而不是等待对模块对象的请求。

现在将参照图 4 (a) 到 (d) 来描述特定情况。

在此情况中，应用程序有对两个对象的打开参考，这两个对象在相同模块中传送。两个对象的版本均为 N。当前可在流上获得的版本为 N+1。然而，此时，应用程序不知道这点。

图 4 (a) 中, 说明应用程序通过版本为 N 的所需模块的 BIOP 消息 1 和 BIOP 消息 2 来参考对象 1 和对象 2。

图 4 (b) 中, 应用程序请求对象 3。此时, 数据轮放层检查可在流上获得的版本, 并发现版本 N+1。

如图 4 (c) 所示, 数据轮放层通知对象轮放层可在流上获得新版本的模块。对象轮放层因此从模块复制其参考的所有 BIOP 消息并关闭所有对它的访问。

现在可安全地装载新版本的模块, 并且如图 4 (d) 所示, 在装载完成时, 对象 3 现在访问最新的版本, 而对象 1 和对象 2 仍访问以前的版本。

图 1

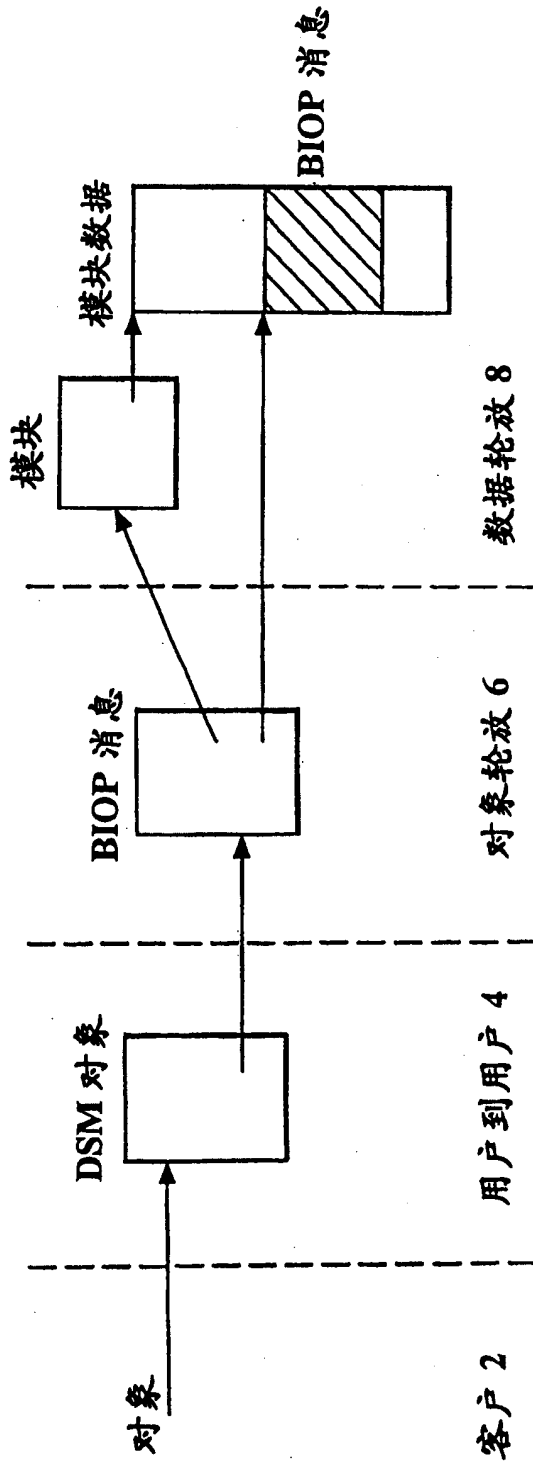


图 2

