

[19] 中华人民共和国国家知识产权局

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



# [12] 发明专利申请公布说明书

[21] 申请号 200580034084. X

[43] 公开日 2007年9月26日

[11] 公开号 CN 101044452A

[22] 申请日 2005.10.5

[21] 申请号 200580034084. X

[30] 优先权

[32] 2004.10.5 [33] GB [31] 0422092.7

[86] 国际申请 PCT/GB2005/003829 2005.10.5

[87] 国际公布 WO2006/038003 英 2006.4.13

[85] 进入国家阶段日期 2007.4.5

[71] 申请人 UIQ 技术公司

地址 瑞典罗奈拜

[72] 发明人 马蒂亚斯·瑞克 约翰·卡内加德  
伊戈尔·内托

[74] 专利代理机构 北京三友知识产权代理有限公司  
代理人 李 辉

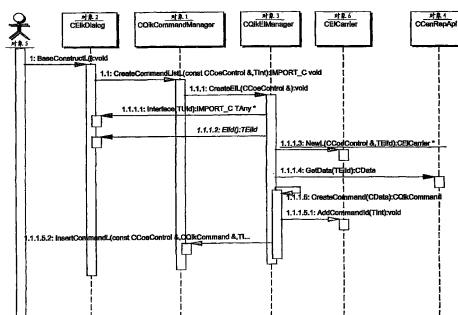
权利要求书 2 页 说明书 11 页 附图 5 页

## [54] 发明名称

计算装置上的应用定制

## [57] 摘要

本发明涉及计算装置上的应用定制。计算装置包括应用框架，该应用框架对用户接口控件和用户输入进行处理并且提供应用所需要的用户接口服务。在可独特标识的控件与可独特标识的任务或服务请求之间建立嵌入式链接，该嵌入式链接存储在装置的数据存储部中。当调用控件时，对该存储部进行检查，以识别出独特地指向该控件的任何嵌入式链接。然后对该控件进行修改，以使得用户或运营商可以附加地选择与附加到所标识出的嵌入式链接的任务或服务有关的选项，并且应用框架在用户或运营商选择已添加到该控件的附加选项的情况下发出所识别出的任务或服务请求。



1、一种计算装置，该计算装置包括：

- a. 应用框架，其可进行操作以提供应用共同需要的用户接口服务；
- b. 所述应用框架可进行操作以处理用户接口控件和用户输入；并且
- c. 可以独特标识应用的用户接口控件；
- d. 可以以任一任意应用的名义请求应用执行特定任务或服务；
- e. 可以独特地标识这种任务或服务请求；并且
- f. 可以在可独特标识的控件与可独特标识的任务或服务请求之间建立嵌入式链接；并且
- g. 可以将这种嵌入式链接保持在所述装置中的数据存储部中；并且其中
- h. 所述应用框架被配置成当调用控件时对所述存储部进行检查，以识别出独特指向该控件的任何嵌入式链接，并对所述控件进行修改以使得用户或运营商可以附加地选择与附加到所识别出的嵌入式链接的任务或服务有关的选项；并且
- i. 所述应用框架被配置成在用户或运营商选择已添加到所述控件的附加选项的情况下，发出所识别出的任务或服务请求。

2、根据权利要求1所述的装置，其中，外部应用或代理可以进行操作，以将可独特标识的控件与可独特标识的任务或服务请求之间的链接添加到所述装置中的保持嵌入式链接的所述数据存储部中。

3、根据权利要求1或2所述的装置，其中，针对未授权访问对所述数据存储部进行保护。

4、一种对计算装置进行操作的方法，该计算装置包括：应用框架，用于提供应用共同需要的用户接口服务并用于处理用户接口控件和用户输入；可以独特标识应用的用户接口控件；可以以任一任意应用的名义请求应用执行特定任务或服务；并且其中，可以独特标识这种任务或服务请求；该方法包括以下步骤：

- a. 在可独特标识的控件与可独特标识的任务或服务请求之间建立嵌

入式链接；和

b. 将这种嵌入式链接保持在所述装置中的数据存储部中；

c. 当调用控件时利用所述应用框架对所述存储部进行检查，识别出独特指向该控件的任何嵌入式链接，并对所述控件进行修改以使得用户或运营商可以附加地选择与附加到所识别出的嵌入式链接的任务或服务有关的选项；以及

d. 在所述用户或运营商选择已添加到所述控件的附加选项的情况下发出所识别出的任务或服务请求。

5、根据权利要求 4 所述的方法，其中，外部应用或代理可以将可独特标识的控件与可独特标识的任务或服务请求之间的链接添加到所述装置中的保持嵌入式链接的永久数据存储部中。

6、根据权利要求 4 或 5 所述的方法，其中，针对未授权访问对所述永久数据存储部进行保护。

7、一种用于使得计算装置根据权利要求 4、5 或 6 中的任何一项所述的方法来进行操作的操作系统。

## 计算装置上的应用定制

### 技术领域

本发明涉及对计算装置上的应用的功能性的生产后定制。

### 背景技术

针对特定客户或特定市场对所制造的产品进行特殊的修改是普遍的。一般来说，除了通常的制造成本以外，一定会产生这种定制的成本，这意味着，如果出现这种定制，则要么产品的价格必须升高要么利润率必定降低。因此进行定制是否有意义取决于由任一方引起的修改的效用是否胜过成本。换句话说，成功的定制作业的关键是在使对制造过程的修改的成本最小化的同时使其有用性最大化。

在价格敏感市场中所需的经济权衡尤为尖锐，在价格敏感市场中，价值链中的经销商都试图使他们自己与他们的竞争对手区分开来。对产品进行修改是增加价值并通过提供更适合于特定使用模式的某些东西来使所提供的产品有所区别的一种显然的方式；但是与此同时，这不能以侵蚀利润率或给予竞争对手价格优势为代价。

在消费电子产品领域中可以发现这些经济力量起作用的某些最明显的示例。这里利用高级移动电话的制造和销售作为本发明可以应用的领域的一个例示；然而，本发明并不是要将本发明的应用只限于移动电话的领域，因为在交互式计算装置的整个范围上都可以通用本发明。

应当将术语“交互式计算装置”解释成包括任何如下装置，即，其既包括用于显示信息的屏幕或其它方法又包括用于输入诸如选择信息的键盘、小键盘、按钮阵列、触摸屏或某些其它方法；现在，这包括：诸如台式计算机、膝上型计算机、PDA、移动电话、智能电话、数字摄像机、数字音乐播放器的个人装置；和许多其它工业和家用装置，其范围涵盖从 ATM 到家用电气设备（如洗涤机和电视）到包括这种装置的诸如

（所有形式的）机动车、火车、船以及飞机的运输机构。

交互式计算装置通常包括可编程软件控件。可以将控件定义为当由用户操纵时产生即时动作或可见结果的对象；通用的控件包括弹出、下拉或选项卡式菜单、单选按钮以及按压按钮、热点以及超链接、复选框、对话框、滚动条以及滑块（slider），而通用的操作装置包括键、按钮以及诸如触摸屏、操纵杆或鼠标的点击装置。

通常由网络运营商（其商业模式依赖于为他们的订户提供服务）来经销移动电话。近年来，网络运营商将这些移动电话产品的定制版代销给他们的订户以有助于区分相应运营商的服务，这已经变得普遍。

然而，为了将产品定制的成本保持在可接受限度内，网络运营商通常只对产品进行表面性的修改。这些修改通常包括装饰性的变更，如对电话的外部塑料应用标识或定做的颜色以强化运营商的品牌。

就对电话进行控制的软件而言，通常所作的独特修改是对已有应用进行预配置或预装专门编写的软件。在本领域中公知的是，修改请求等同于产品规格的新近改变，并且这种改变请求是在投放市场之前和在出货之后引起产品故障的主要原因。

因此软件开发经济趋于使得对移动电话和其它交互式消费电子装置中的嵌入式应用的任何更深入或更基础的定制惊人地昂贵；制造商和经销商意识到，复杂软件在架构上是脆弱的，并且从时间和人工两方面来说所需的重复编译和测试周期是昂贵的。

经济性的实践是只对复杂软件产品进行表面性的修改，该事实已经限制了定制性能的潜在有用性。例如，在移动电话领域，如果可以通过某种方式将修改嵌入在已有应用中而不是借助于附加应用来覆写修改，那么这显然对于用户和网络运营商来说都是极其有用的。

该定制性能的一些示例有：

- 对电话应用功能性进行修改以添加用以呼叫某个运营商专用服务号码的菜单项
- 对浏览器应用功能性进行修改以添加用以连接到运营商专用网站并且可能下载材料的菜单项

- 对消息传送应用进行修改以添加用以向客户关系地址发送电子邮件的选项。

此外，如果不仅可以在制造之后尽可能晚地进行这种嵌入式应用修改，而且可以在已销售这些装置之后（即，当端用户拥有它们时）在它们中实现这种嵌入式应用修改，那么这将是极其有用的。利用嵌入到每个电话中的无线连接来进行的无线电式（OTA）定制将是借以实现上述目的的理想机制。

用于对应用进行修改的已知方式只可以单独应用于这些缺陷。

通过使已有应用利用父—子范例产生可执行程序，可以将修改嵌入在这些应用内（并改变这些应用）。可以这样做，这是大多数操作系统（包括使移动电话运行的操作系统）的公知性质；显然，如果将应用编写成在其执行序列中的特定点处对这种可产生程序（通常称为插件）进行搜索，那么对插件的更改不需要修改该应用或操作系统的低层架构。

此外，用于通过按无线电方式发送这种插件来在适当位置更改它们的技术已经存在一段时间，并且用户通常已利用该技术的各种形式来下载铃声和游戏，并且网络运营商已利用该技术的各种形式来向新订户发送配置数据和临时定做功能性。

然而，这些已知方法是不足的，因为它们是不可扩展的；所描述的这些技术要求网络运营商（或试图对内嵌应用进行定制的其他代理商）预先了解可能需要添加定制功能性的号码和可能的位置，并且要求制造商在制造装置时在应用中放置合适的钩子（hook）。如果在制造时未设置钩子，则由应用执行所下载的插件或对其菜单和对话框进行定制以表示在哪儿设置了附加功能性是不可能的。

因此，在不同方对哪儿可能需要钩子具有不同观点的情况下，制造商要么必须支持所有方的要求，要么必须预先决定哪些方的要求是最重要的。这严重地限制了可能的定制的效用。通过添加多个钩子来支持所有可能的定制几乎一定会被证明不仅在架构上是复杂的而且惊人地昂贵。

此外，利用公知的定制技术不可能在制造后阶段向装置添加未预料

到的定制性能。

尽管可以改变特定定做项目，但是不能容易地增加它们的数量或移动它们位置。

#### 发明内容

因此本发明的一个目的是提供一种对计算装置上的应用进行定制的改进方式。

根据本发明的第一方面，提供了一种计算装置，该计算装置包括：

- a. 应用框架，其可进行操作以提供应用共同需要的用户接口（UI）服务；
- b. 所述应用框架可进行操作以处理 UI 控件和用户输入；并且
- c. 可以独特标识应用的 UI 控件；
- d. 可以以任一任意应用的名义请求应用执行特定任务或服务；
- e. 可以独特标识这种任务或服务请求；并且
- f. 可以在可独特标识的控件与可独特标识的任务或服务请求之间建立嵌入式链接；并且
- g. 可以将这种嵌入式链接保持在所述装置中的数据存储部中；并且其中
- h. 所述应用框架被配置成当调用控件时对所述存储部进行检查，以识别出独特指向该控件的任何嵌入式链接，并对所述控件进行修改以使得用户或运营商可以附加地选择与附加到所识别出的嵌入式链接的任务或服务有关的选项；并且
- i. 所述应用框架被配置成在用户或运营商选择已添加到所述控件的附加选项的情况下，发出所识别出的任务或服务请求。

根据本发明的第二方面，提供了一种对计算装置进行操作的方法，该计算装置包括：应用框架，用于提供应用共同需要的用户接口服务并用于处理 UI 控件和用户输入；可以独特标识应用的 UI 控件；可以以任一任意应用的名义请求应用执行特定任务或服务；并且其中，可以独特地标识这种任务或服务请求；该方法包括以下步骤：

- a. 在可独特标识的控件与可独特标识的任务或服务请求之间建立嵌入式链接；和
- b. 将这种嵌入式链接保持在所述装置中的数据存储部中；
- c. 当调用控件时利用所述应用框架对所述存储部进行检查，识别出独特指向该控件的任何嵌入式链接，并对所述控件进行修改以使得用户或运营商可以附加地选择与附加到所识别出的嵌入式链接的任务或服务有关的选项；以及
- d. 在所述用户或运营商选择已添加到所述控件的附加选项的情况下发出所识别出的任务或服务请求。

根据本发明的第三方面，提供了一种用于使得根据第一方面的计算装置根据第二方面的方法来进行操作的操作系统。

#### 附图说明

以下将参照附图，仅通过进一步示例的方式对本发明的实施例进行描述，在附图中：

图 1 示出了根据本发明的具有可定制元素的应用框架的组成部分概貌；

图 2 示出了图 1 所例示的框架的典型嵌入式链接（EL）命令详情；

图 3 示出了可以如何创建嵌入式链接（EL）；

图 4 示出了可以如何对计算装置的操作系统的中央仓库对象进行更新；以及

图 5 示出了对 UI 项目的析构。

#### 具体实施方式

本发明可应用于并入了如下操作系统层的任何计算装置：该操作系统层提供用于处理所有交互式应用共有的用户接口（UI）元素的编程框架。在这种框架中包括的应用编程接口（API）（尤其）实现以下任务中的一些或全部：

- 用户接口控件的创建

- 针对用以处理用户接口事件的应用的框架
- 针对控件创建的环境实用程序
- 对开窗（windowing）功能性的访问
- 处理不同形式的用户输入。

在提供图形用户接口的所有当前操作系统中通常都提供了这种应用框架（AF）；它消除了所有应用都要包括用于它们自己处理这种任务的单独代码的必要性，并且还使得所有应用都能够协同地共享具有共同外观和感觉的 UI。

在本发明中，不是将可定制元素直接嵌入在已有应用中，而是完全在应用外部对它进行定义。在本发明的语境下将这种可定制元素称为外部定义命令（EDC），并且可以由 AF 将该可定制元素自动地添加到任何应用的调用该可定制元素的任何位置处。EDC 可以产生可执行程序的新实例并向该新实例传递必要的指令，或者，如果该可执行程序已在运行，则可以利用进程间通信来传递请求。在任一情况下，都将应用的控件与 EDC 之间的联接称为嵌入式链接（EL）。

根据本发明，独特地标识每个应用控件，而且独特地标识每个 EL。这使得可以构造出将这些嵌入式链接映射到这些控件的数据库；具体的机制可以随实现而不同，但是向该数据库写元组是一种可能性。

一旦构造了该数据库，则每当调用 AF 来创建控件（如菜单）时就将 AF 配置成查询该数据库。如果针对某个控件找到了 EL，则此时利用该时机对应用的控件进行定制，而操作系统或该应用的生产商不必预先提供任何特殊的钩子。AF 只需检查是否存在对应于该控件的嵌入式链接。

存在可以对控件进行定制的许多可能的机制；由于嵌入式链接的编写者必须知道需要对什么类型的控件进行定制，因此简化了该任务。最简单的机制是将嵌入式链接从元组扩展成三元组，第三个元素包含有定制；因此在本实施例中与菜单控件相关联的三元组包括可以添加到菜单中的某些文本。

一旦 EL 在所显示的控件中表现为选项，那么只要也利用 AF 来接受

输入，就会调用正确的 EDC。

该机制的可扩展性源自可以对将控件映射到 EL 的数据库进行写和读的事实。在理想的情况下，该实现应当基于安全的操作系统（OS），在安全的 OS 中，将对数据库进行写的能力限于具有合适许可的系统应用。由英格兰伦敦的 Symbian 软件有限公司提供的在英国专利申请 0312191.0（Capabilities）和 0312190.2（Data Caging）中描述的“平台安全性架构”是这种 OS 的一个示例。

利用该发明，应用编写者不必预先知道可能要发生定制。唯一的限制是：定义 EL 的一方必需确保在定义 EL 之前知道应当向其附加 EDC 的应用和具体控件。

以下示例示出了针对由瑞典罗奈拜的 UIQ 技术公司为 Symbian 软件有限公司的高级操作系统 Symbian OS 编写的可定制 UIQ 用户接口而开发的优选实现，给出该示例是为了例示本发明背后的原理，而不应当将该示例视为表示本发明的可应用性在任何方面局限于该特定操作系统和用户接口。

因此尽管对 Symbian OS 和 UIQ 的一定程度的熟悉对于全面理解该具体实现来说是必要的，但是如所公开的那样的本发明背后的原理将适用于具有应用框架和用于调用插件应用的机制的任何 OS。设计用户接口和操作系统的领域内的任何技术人员都很容易理解这里所公开的技术内幕。

该具体实现的一些显著特征如下：

- Symbian OS 包括在每个相继的 OS 版本中都得到了不断改进的 AF（有时称为 CONE，控件环境的缩写）。这提供了用于创建用户接口控件的框架、应用用以处理用户接口事件的框架、以及用于进行控件创建和访问开窗功能性的环境实用程序。
- Symbian OS 还可以包括中央仓库，在此情况下 Symbian OS 充当将 EL 映射到独特控件的数据库。
- 在 UIQ 用户接口中，通过被称为直接导航链接（DNL）的机制来实现 EDC。使用 DNL 来从一个应用直接导航到另一个应用，

这通常是为了使得装置的用户尽可能容易地执行任务。DNL 通常在特定环境下起作用，从一个应用获取数据并将该数据传递给另一个应用以使得可以直接使用该数据。例如，在联系人应用中使用 DNL，使得用户可以通过敲电话号码来直接发起电话呼叫或通过敲电子邮件地址来创建电子邮件。

- UIQ 还包括命令处理框架 (CPF)，在英国专利申请第 0414842.5 中已描述了 CPF；在该实现中，嵌入式链接从中央仓库中读取信息，并被添加到 CPF 中的合适命令列表中。

该机制不仅可以用于定义 EL，而且可以用于定义某些种类的系统命令、被添加到任何应用的命令（如“挂断电话呼叫”）。在此情况下，不必提前知道该应用。此外，可以对该机制进行扩展以允许某些种类的应用交互。应用交互轻微地违背了应用不知晓添加的原理，但是这使得可以利用某些环境信息来对链接进行定制。作为示例，当用户在“日程”中激活 EDC 时，可以显示正确的日期的天气信息。

另一场景是，在 GPS 应用的情况下，可以提供与当前位置有关的更详细的信息。

因此本发明提供了这样一种方法：其通过利用所有的高级装置依赖于应用框架来处理公共 UI 控件和输入的事实，结合一个应用向另一应用（根据底层 OS，可能附加地生成第二应用）传递请求的能力，对计算装置上的应用进行定制。只有应用框架可以独特地标识特定控件并且可以独特地标识应用请求，就可以由链接到特定请求的特定控件来构造数据库。当调用控件时由应用框架来查询具有嵌入式链接的该数据库，并且如果存在独特指向该受关注控件的相关嵌入式链接，则应用框架可以对该控件进行修改以添加针对所链接的请求的选项。对装置的任何部分的定制于是成为仅仅对数据库进行写入的事情。

图 1 示出了具有可定制元素的应用框架的组成部分概貌，图 2 示出了典型的 EL 命令详情。

可以使用以下类来实现嵌入式链接和外部定义命令。本领域的技术人员很容易理解这些类。

### CQikElManager

CQikElManager 的责任是跟踪所有 EL 载体并从中央仓库中读取信息，然后将它们添加到 CPF 中的合适命令列表中。

当调用 CreateEIL()时，对 CCoeControl 的对象供应器接口进行调用以获得指向 MEIInterface 类的指针。该类由可以包含 EL 的所有 UI 项目来实现。调用 EIID()以得到 UI 项目的 EL id。命令处理器将使用该 id 来从 CenRep 中找出 EL 信息。从该 EL 信息中读取 EL 类型，然后创建合适的 CQikElCommand 实例化，并将该 CQikElCommand 实例化添加到该 UI 项目的命令列表中。

### CElCarrier

CElCarrier 保持与包含有 EL 的各 UI 项目有关的信息。该信息由 TEIID 和与 UI 项目相关联的 EL 的命令 id 的列表组成。

### CQikElCommand

CQikElCommand 是所有 EL 命令类型的基类。它保持与 EL 类型有关的信息。将指向其 MQikCommandHandler 的指针设定为指向命令本身。

### CQikUrlEl

CQikUrlEl 实现 URL EL。当调用 HandleCommandL()时，产生到内部浏览器应用（QWebExternalInterface.h）的 DNL。该 DNL 包含待由浏览器打开的 URL。

### CQikSmsEl

CQikSmsEl 实现 SMS EL。当调用 HandleCommandL()时，产生到消息传送应用的 DNL。该 DNL 包含有 CSendAs 对象，该 CSendAs 对象又由接受者号码和 SMS 消息体组成。

### CQikMmsEl

CQikMmsEl 实现 MMS EL。当调用 HandleCommandL()时，产生到消息传送应用的 DNL。该 DNL 包含有 CSendAs 对象，该 CSendAs 对象又由接受者号码和 MMS 消息体组成。

### CQikEmailEl

CQikEmailEl 实现 Email EL。当调用 HandleCommandL()时，产生到

电子邮件应用的 DNL。该 DNL 包含有 CSendAs 对象，该 CSendAs 对象又由接受者电子邮件地址和电子邮件消息体组成。

### CQikCallEl

CQikCallEl 实现呼叫 EL。当调用 HandleCommandL()时，产生到电话应用的 DNL。该 DNL 包含有待呼叫的电话号码。

### CCenRepApi

CCenRepApi 对中央仓库的接口进行封装。在决定 CenRep 的 API 之前，该类的详情是 TBD。对于 MCenRepObserver 也是如此。

### MEIInterface

MEIInterface 是当请求针对 EL 载体的接口时 CCoeControl 的对象供应者接口返回的类。该类必须由可以包含 EL 的所有 UI 项目来实现。

### TEIID

TEIID 利用两个标识符 (UID) (一个应用 UID 和一个 EL UID) 来独特地标识可以包含 EL 的 UI 项目。为了处理不属于应用的项目，在应用 UID 数据库中注册一系统 UID。在应用 UID 的范围内该 EL UID 必须是独特的。

可以修改 UIQ 和 Symbian OS 中的以下类来实现本发明：

### CQikCommandManager::TManagerData

添加新成员 CQikElManager 指针。

### CQikCommandManager

在 ConstructL() 中添加对 CQikElManager 对象的创建和对 TManagerData 中的 CQikElManager 指针的赋值。

在 CreateCommandListL() 中添加对 CQikElManager.CreateEIL() 的调用。

在 ResetAndDestroyResourcesFor() 中添加对 CQikElManager.RemoveEl() 的调用。

### CEikDialog

实现 MEIInterface 类和 MObjectProvider 支持。

### CQikViewBase

实现 MEIInterface 类和 MObjectProvider 支持。

为了标识 UI 项目，可以对 3 个不同的 UI 项目视图、对话框以及弹出体（pop out）指配 EL。最终可以添加附加的 UI 项目。要包含 EL 的每个 UI 项目都应当是可独特地标识的。使用 TEIId 标识符来标识它们。

TEIId 的前缀部分是应用 UID。TEIId 的后缀部分是 EL UID。在应用 UID 的范围内 EI UID 必须是独特的。存在实现该目的的许多不同的方式。

一个选项是使用针对视图的视图 id。对该解决方案的一个顾虑是对话框和弹出体都没有视图 id。因此，如果要采用该选项，那么必须为对话框和弹出体连同视图一起分配视图 id，但是这存在将它们混合起来可能会产生混淆的附加缺点。

另一选项是使用资源 id。该选项的一个优点是，许多 UI 项目是从资源文件创建的，因此具有资源 id。一个缺点是通常视图不是从资源文件创建的。弹出体也没有资源 id。再一缺点是，资源 id 必须在时间上并且在多次编译之间是一致的。这要求当改变资源文件时将二进制兼容性问题考虑在内：资源 id 的一致性取决于不从资源文件擦除资源名称。

第三选项是使用 CCoeControl 中的独特 id。该选项的一个优点是，可以包含有 EL 的所有 UI 项目都是 CCoeControl 的子代，因此具有独特 id。缺点是并非全都具有独特 id 组。这可能意味着需要对许多 UI 项目进行更新以使得它们能够包含 EL。不存在对当创建命令列表时的独特 id 变化的支持。可以通过所需要的有限努力来添加该功能性。因此，该第三选项是用以建立独特 id 的优选选项。图 3 示出了可以如何创建 EL。

图 4 示出了可以如何对中央仓库（CenRep）进行更新，图 5 示出了可以如何消灭 UI 项目。

尽管已参照特定实施例对本发明进行了描述，但是应当明白，在保持在如所附权利要求所限定的本发明的范围之内的同时，可以实施修改。

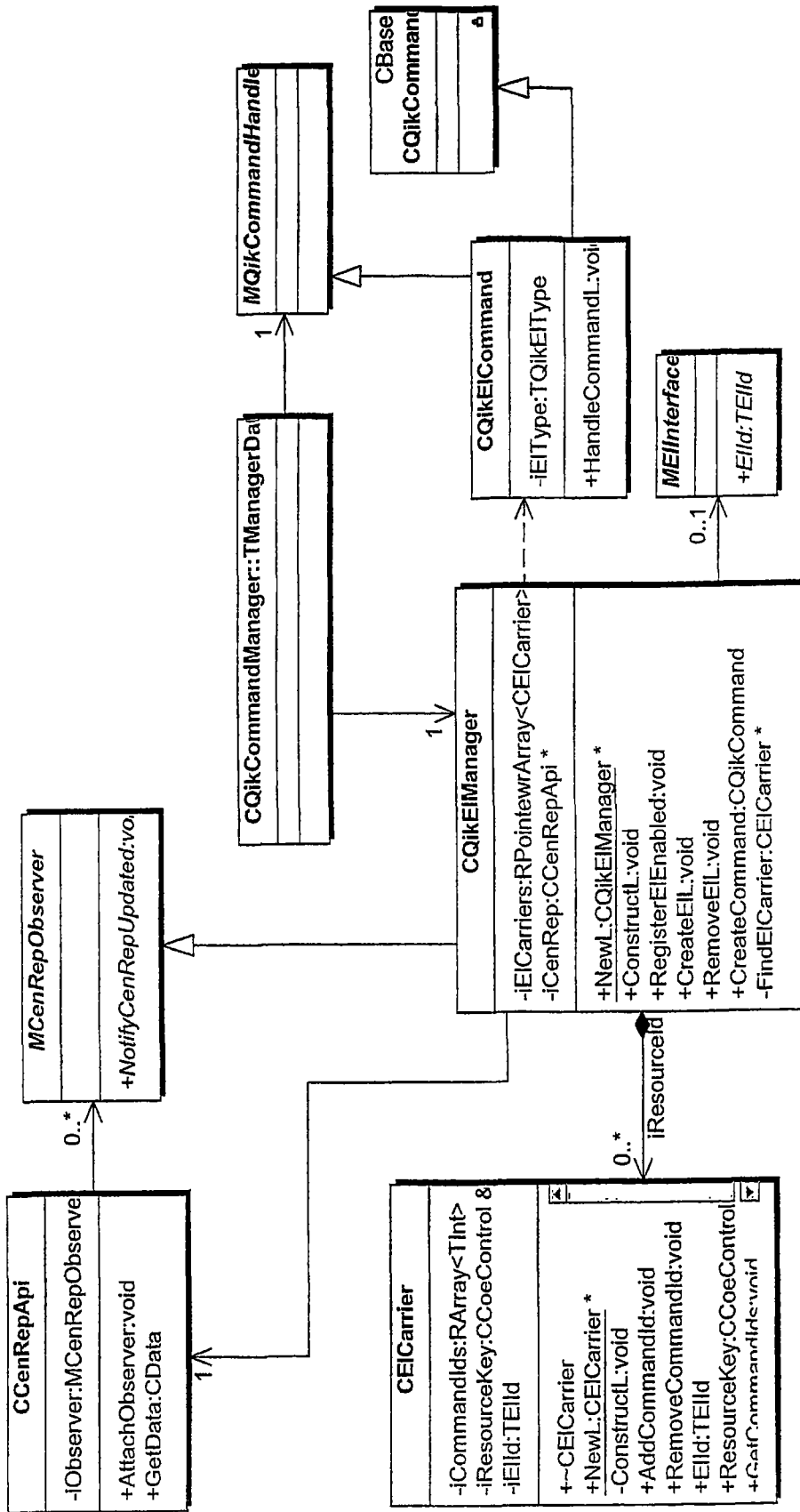


图 1

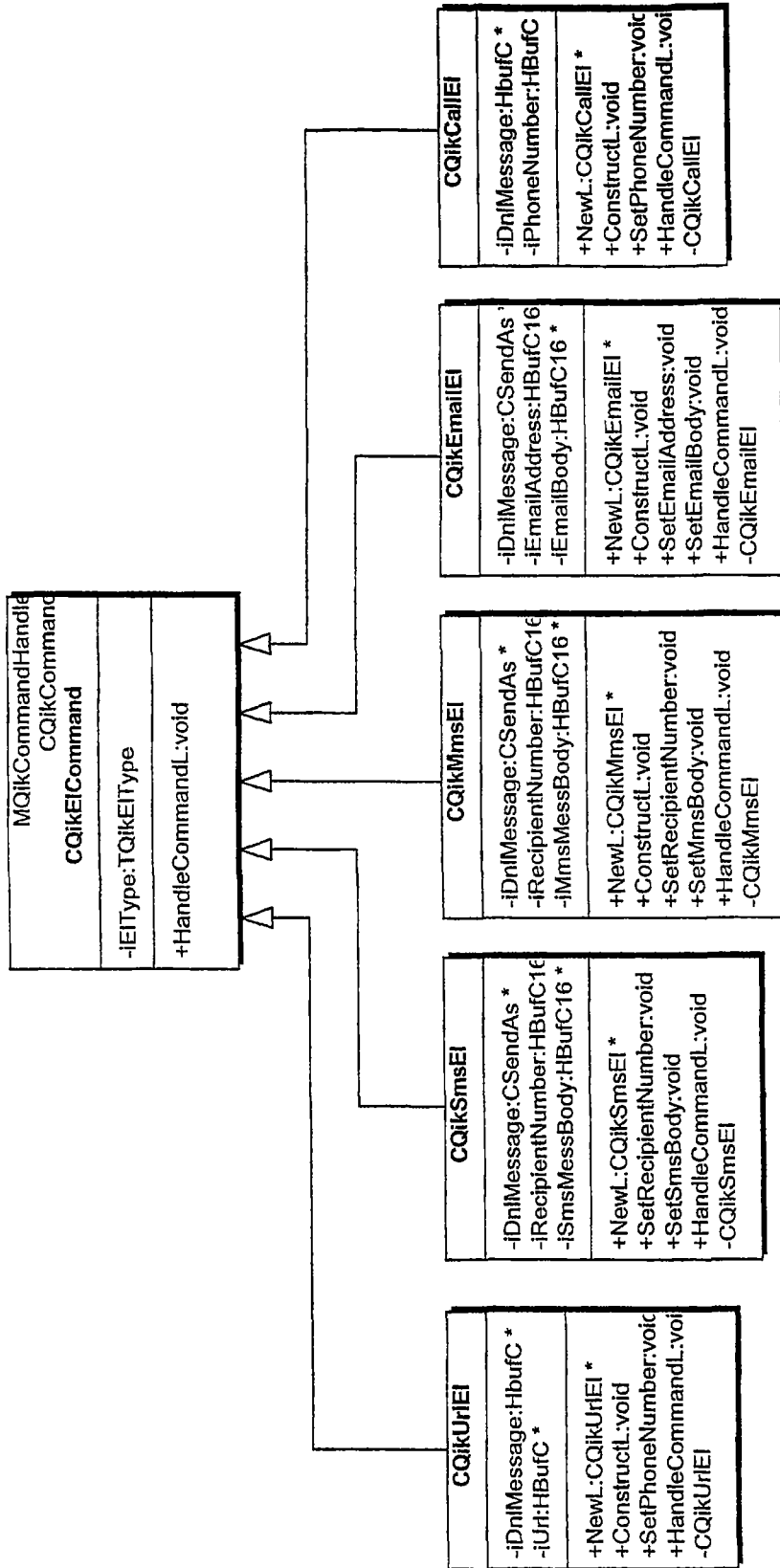


图 2

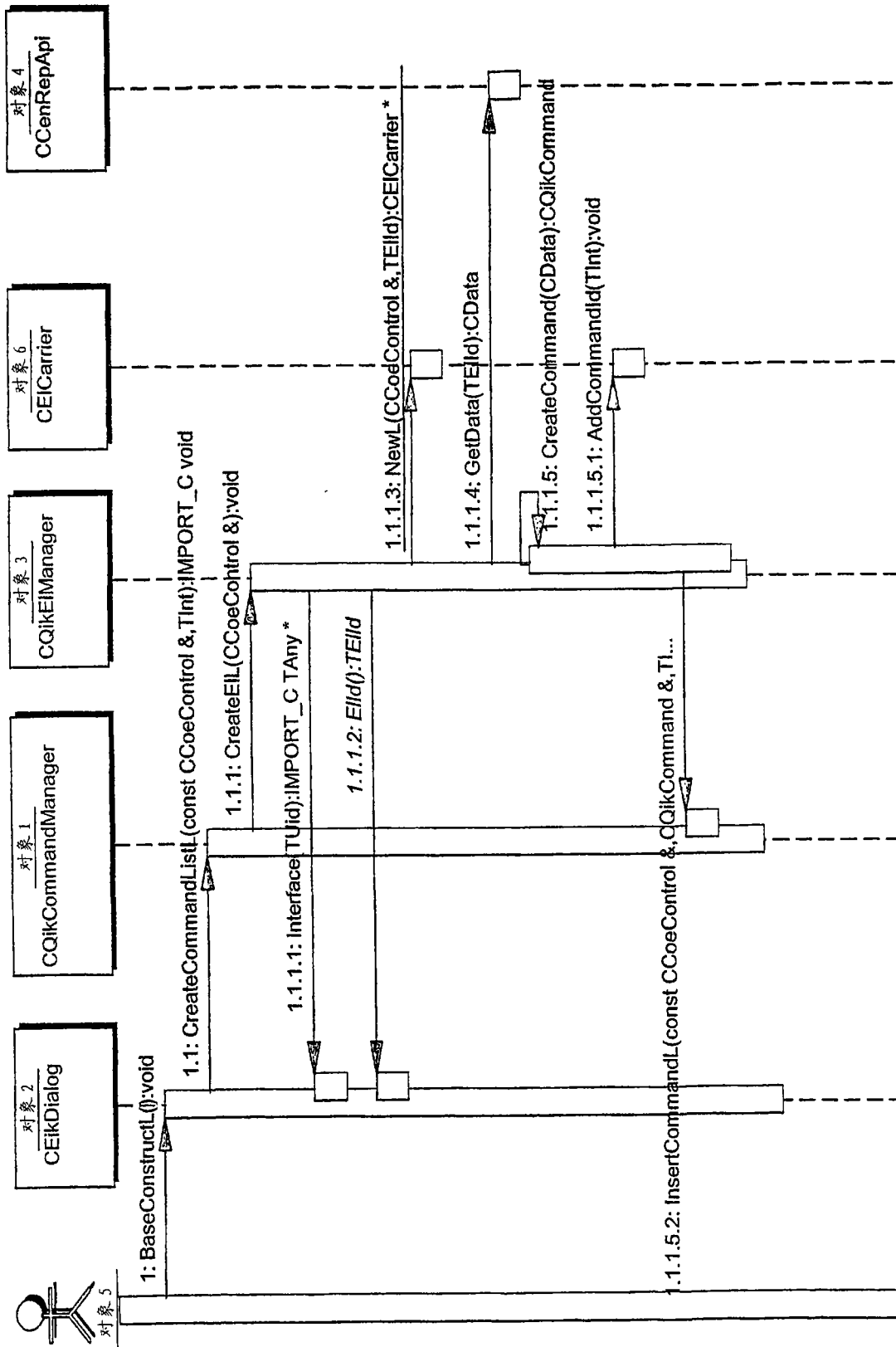


图 3

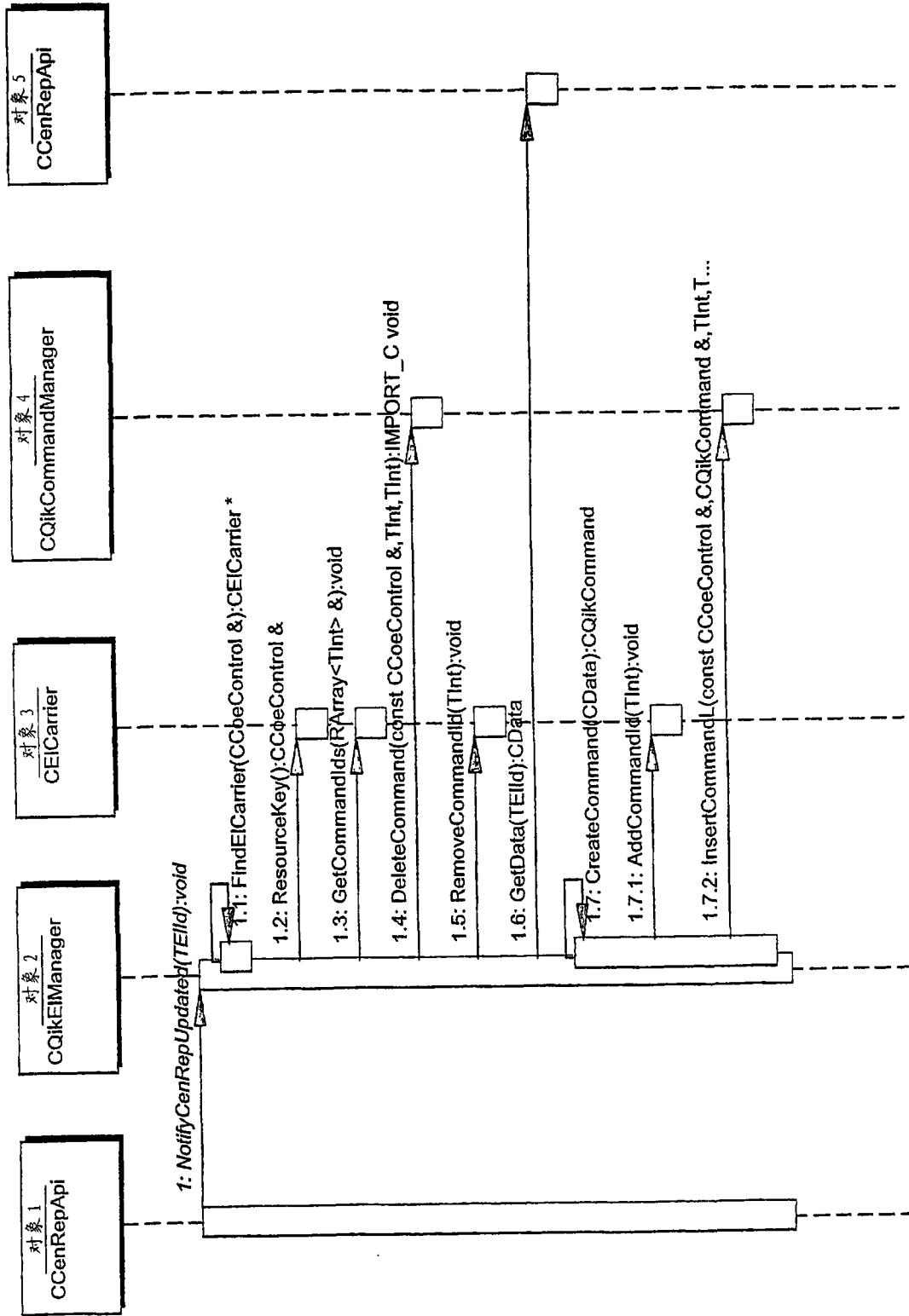


图 4

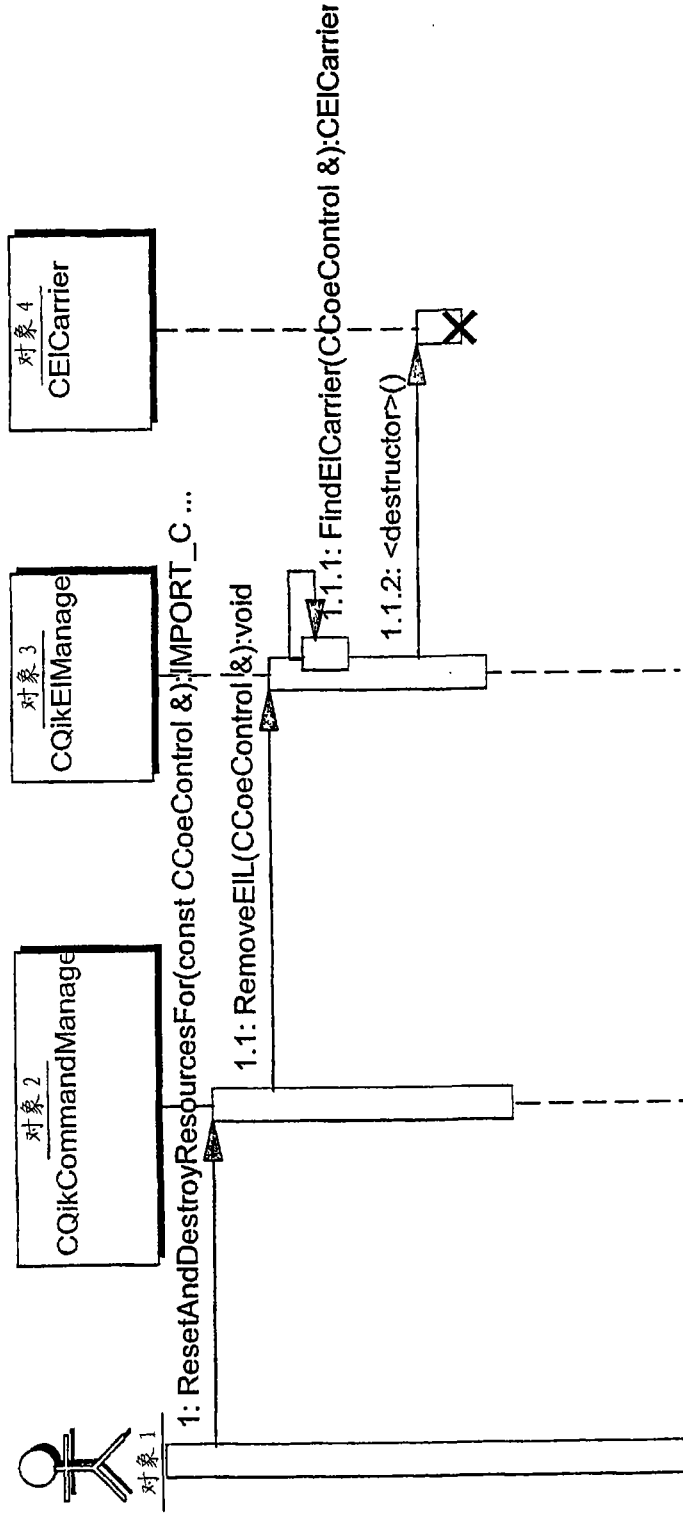


图 5