

[19]中华人民共和国专利局

[51]Int.Cl⁶

G06F 9/44

H04Q 3/545



[12] 发明专利申请公开说明书

[21] 申请号 95193593.3

[43]公开日 1997 年 5 月 28 日

[11] 公开号 CN 1150844A

[22]申请日 95.6.7

[30]优先权

[32]94.6.14 [33]US[31]08/259,785

[86]国际申请 PCT/SE95/00666 95.6.7

[87]国际公布 WO95/34856 英 95.12.21

[85]进入国家阶段日期 96.12.13

[71]申请人 LM爱立信电话有限公司

地址 瑞典斯德哥尔摩

[72]发明人 富莱吉克·斯范菲尔德

[74]专利代理机构 中国国际贸易促进委员会专利商标
事务所

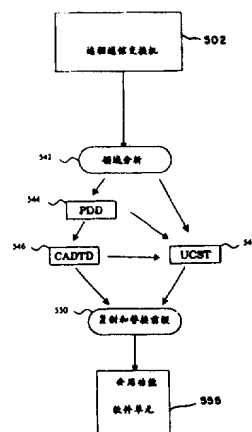
代理人 鄢 迅

权利要求书 3 页 说明书 9 页 附图页数 6 页

[54]发明名称 为远程通信交换机建立软件的方法

[57]摘要

本发明是一种建立适于控制远程通信交换机的公用功能性软件单元(555)的方法。公用功能性软件单元(555)是通过对所有现有远程通信交换机(502)进行领域分析(542)、然后对交换机的承载业务处理程序(BSH)和硬件类型的所有可能组合的公用概念、术语、行为和功能进行标识而得以建立的。公用行为和功能是利用通用公用软件模板(UCST)(548)、相关联的参数定义文档(PDD)(546)和相关联的抽象数据类型定义集合(CADTD)(546)实现的。对UCST进行编译并用CADTD的子集进行测试。在编译和测试UCST之后,可以从UCST和CADTD建立BSH和HW的不同组合的多个公用软件功能单元(CS)。



(BJ)第 1456 号

权 利 要 求 书

1.为多个远程通信交换机中的一个指定的交换机建立公用功能性软件单元的方法，所述多个远程通信交换机中的每个具有特定硬件类型并技术特定的承载业务处理程序，所述方法包括下述步骤：

建立包括多个公用功能性软件单元的通用公用软件模板，所述多个公用功能性软件单元中的每一个具有多个模板前缀和至少一个参数；

建立一个抽象数据类型定义集合，所述集合包括每个所述参数的数据类型定义；

从所述通用公用软件模板复制所述多个公用功能性软件单元中的一个，所述所复制的公用功能性软件单元与所述指定的远程通信交换机的所述特定承载业务处理程序和所述特定硬件类型兼容；以及

用标识所述指定的远程通信交换机的所述特定承载业务处理程序和所述特定硬件类型的所选择的前缀替换所述所复制的公用功能性软件单元中的多个模板前缀。

2.权利要求 1 的方法，还包括步骤：

建立包括每个所述参数的至少一个定义的参数定义文档。

3.权利要求 1 的方法，其中所述建立通用公用软件模板的步骤包括步骤：

标识多个组合，所述多个组合中的每一个包括所述多个承载业务处理程序之一和所述多个硬件类型之一；

标识存在于所述多个组合的每一个中的多个公用功能；以及

建立多个类，以实现所述多个公用功能的每一个；

4.权利要求 3 的方法，其中所述建立多个类的步骤包括步骤：

将一个模板前缀附加到所述多个类中的每一个。

5.权利要求 4 的方法，其中所述建立抽象数据类型定义集合的步骤包括步骤：

将一个独立的模板前缀附加到所述数据类型定义的每一个，所述独立模板前缀独立于所述多个承载业务处理程序和所述多个硬件类型。

6.权利要求 5 的方法，其中所述建立抽象数据类型定义集合的步骤

包括步骤:

将第一从属模板前缀附加到所述数据类型定义的每一个, 所述第一从属模板前缀依赖于所述多个承载业务处理程序之一, 并独立于所述多个硬件类型。

7. 权利要求 6 的方法, 其中所述建立抽象数据类型定义集合的步骤包括步骤:

将第二从属模板前缀附加到所述数据类型定义的每一个, 所述第二从属模板前缀依赖于所述多个硬件类型之一, 并独立于所述多个承载业务处理程序。

8. 权利要求 7 的方法, 其中所述建立抽象数据类型定义集合的步骤包括步骤:

将第三从属模板前缀附加到所述数据类型定义的每一个, 所述第三从属模板前缀依赖于所述多个组合之一。

9. 权利要求 6 的方法, 其中所述替换所述所复制的公用功能性软件单元中的多个模板前缀的步骤包括步骤:

用标识所述指定交换机的所述特定承载业务处理程序的第一所选前缀替换所述第一从属模板前缀中的每一个。

10. 权利要求 7 的方法, 其中所述替换所述所复制的公用功能性软件单元中的多个模板前缀的步骤包括步骤:

用标识所述指定交换机的所述特定硬件类型的第二所选前缀替换所述第二从属模板前缀中的每一个。

11. 权利要求 8 的方法, 其中所述替换所述所复制的公用功能性软件单元中的多个模板前缀的步骤包括步骤:

用标识所述指定交换机的所述特定硬件类型和所述特定承载业务处理程序的第三所选前缀替换所述第三从属模板前缀中的每一个。

12. 为多个远程通信交换机建立通用公用软件模板的方法, 所述多个远程通信交换机中的每一个具有特定硬件类型并支持特定承载业务处理程序, 所述方法包括步骤:

标识多个组合, 所述多个组合中的每一个包括所述多个承载业务处理程序之一和所述多个硬件类型之一;

标识存在于所述多个组合的每一个中的多个公用功能；以及建立多个类，以实现所述多个公用功能的每一个。

13.权利要求 12 的方法，其中所述建立多个类的步骤包括步骤：将一个模板前缀附加到所述多个类中的每一个。

14.权利要求 13 的方法，其中所述多个类中的每一个包括至少一个参数。

15.权利要求 14 的方法，其中所述建立多个类的步骤包括步骤：建立抽象数据类型定义集合，所述集合包括每个所述参数的数据类型定义。

16.权利要求 15 的方法，其中所述建立抽象数据类型定义集合的步骤包括步骤：

将一个独立的模板前缀附加到所述数据类型定义的每一个，所述独立模板前缀独立于所述多个承载业务处理程序和所述多个硬件类型。

17.权利要求 16 的方法，其中所述建立抽象数据类型定义集合的步骤包括步骤：

将第一从属模板前缀附加到所述数据类型定义的每一个，所述第一从属模板前缀依赖于所述多个承载业务处理程序之一，并独立于所述多个硬件类型。

18.权利要求 17 的方法，其中所述建立抽象数据类型定义集合的步骤包括步骤：

将第二从属模板前缀附加到所述数据类型定义的每一个，所述第二从属模板前缀依赖于所述多个硬件类型之一，并独立于所述多个承载业务处理程序。

19.权利要求 18 的方法，其中所述建立抽象数据类型定义集合的步骤包括步骤：

将第三从属模板前缀附加到所述数据类型定义的每一个，所述第三从属模板前缀依赖于所述多个组合之一。

说明书

为远程通信交换机建立软件的方法

本发明涉及建立用于控制复杂电子系统的软件的方法，更具体地涉及建立用于控制远程通信交换机的软件的方法。

图 1（现有技术）是大致说明在远程通信系统中所使用的现代远程通信交换机 102 的框图。交换机 102 包括至少一个交换机软件模块 110，交换机硬件（HW）108，多个到交换机 102 的数据输入端 104，以及多个来自交换机 102 的数据输出端 106。交换机硬件 108 允许以非常高的数据速率传送数据。

交换机软件模块 110 向交换机用户软件 100 提供交换机 102 的逻辑视图。逻辑视图向交换机用户提供多个格式（每个格式指示如何在交换机 102 中接收和发送数据），以及连接类型（例如，点 - 对 - 点，点 - 对 - 多点，多点 - 对 - 点，或者多点 - 对 - 多点）。后文中，交换机的逻辑视图称为承载业务处理程序（BSH）。BSH 也使得具有交换机硬件 108 有限知识的交换机用户能够控制交换机硬件 108。

在数据输入端 104 以一种特定格式接收输入数据，并且在数据输出端 106 以另一特定格式发送数据。交换机用户软件 100 经过用户接口 112 与交换机 102 通信。交换机用户软件 100 指定：将在一个或多个数据输入端 104 以第一种特定格式接收的数据交换到一个或多个数据输出端 106，并且以第二种特定格式发送。特定格式的某些例子是异步传输模式 - 虚拟通路（ATM - VP），异步传输模式 - 虚拟通道（ATM - VC），同步传输模式 - 64 千位每秒（STM - 64k），和同步传输模式 - 2 兆位每秒（STM - 2M）。

ATM 是基于固定长度 53 - 字节单元的单元 - 交换技术。所有宽带传输，不管是音频、数据、图象还是视频，都被分为单元系列，并经过包括由 ATM 交换机连接的链路的 ATM 网络传送。每条 ATM 链路包括一恒定流的 ATM 单元间隙，在间隙中放入传输，或者如果未被使用则闲

置。ATM 最显著的优点是它均匀地处理业务, 允许一个网络满足许多宽带业务的需要。考虑到 ATM 对传输和交换两者的灵活性和适用性, 1988 年国际电话电报咨询委员会 (CCITT) 国际标准化组织选定它来实现宽带综合业务数字网络 (B - ISDN)。在 ATM - VP 中, 交换是通过利用虚拟通路标识符 (VPI) 值来实现的。相反, ATM - VC 交换是通过利用 VPI 值和虚拟通道标识符 (VCI) 值两者来实现的。

STM 是依赖于对于某个基准以比如帧模式这样的规则的和固定的模式发生的信息的传输和交换方法。

远程通信交换机是根据交换机硬件所支持的交换格式的类型而分类的。例如, 支持 STM 格式的交换机称为 “ STM 交换机”, 而支持 ATM 格式的交换机称为 “ ATM 交换机”。在现代远程通信业中, 交换机被建立来支持多种特定的格式。例如, ATM 交换机可以既支持 ATM - VP 特定格式也支持 ATM - VC 特定格式, STM 交换机既支持 STM - 64k 也支持 STM - 2M。

图 2 (现有技术) 是显示具有交换机软件和交换机硬件的远程通信交换机 306 的框图。交换 306 包括类型 A 的交换机硬件 (SH - A) 314, 和交换机软件单元 (SS1) 308、(SS2) 310 和 (SS - 1 - 2 - A) 312。交换机 306 也支持类型 1 的 BSH (BSH1) 和类型 2 的 BSH (BSH2) (未示出)。软件单元 SS1 308 和 SS2 310 分别依赖于 BSH1 和 BSH2。软件单元 SS - 1 - 2 - A 312 依赖于 BSH1、BSH2 和 SH - A。用户接口 UI - BSH1 302 和 UI - BSH2 304 分别支持 BSH1 和 BSH2, 并且向用户提供在交换机 306 上选择 BSH1 或 BSH2 的能力。

在现代远程通信业中, 软件单元 SS1 308、SS2 310 和 SS - 1 - 2 - A 312 是仅对在具有特定类型的硬件且实现至少一种 BSH 的交换机中的一种交换机 (ATM 或 STM) 上进行操作而建立的。换言之, 软件单元 SS1 308、SS2 310 和 SS - 1 - 2 - A 312 不是从通用公用软件模板 (UCST) 建立的。

对于远程通信交换机的软件单元的当前设计有几点不利。一种不利来自于对依赖于具体类型的 BSH 或交换机硬件的每个软件单元的更新或修改。例如, 对交换机硬件的修改必须反映到对依赖于交换机硬件的每

个软件单元的更新，而且对具体 BSH 的修改必须反映到对依赖于 BSH 的每个软件单元的更新。当前软件单元设计的另一不利来自于不同组的开发人员建立每种类型交换机的软件单元。在建立软件单元期间，不同组的开发人员经常使用不一致的术语、概念和结构。软件单元的非一致性造成在设计软件单元接口的交换机用户软件时增加了困难。

具有能够建立对于多种类型的交换机硬件和 BSH 的公用软件单元的通用公用软件模板（UCST）有显著优点。本发明提供这样一种 UCST。

本发明的一个目的在于提供一种不需要重复更新或修改的能够设计和建立公用功能性软件单元的通用公用软件模板（UCST）。

本发明的另一目的在于建立自动反映对 UCST 的任何修改的公用功能性软件单元。

本发明仍有一个目的在于降低设计、建立、维护和更新公用功能性软件单元所用的时间和费用。时间和费用的降低是由于设计人员只需熟悉 UCST 而不需熟悉各种公用功能性软件单元。

本发明还有一个目的在于将软件文档减少到用于实现所有公用功能性软件单元的单个集合的文档。单个集合的文档避免了术语和短语使用中的不一致和混乱。

参照附图，结合相应说明，将更好地理解本发明，并且其多个目的和优点对于熟练的技术人员将会更加明显。

图 1（现有技术）是说明在远程通信系统中使用的现代远程通信交换机 102 的框图；

图 2（现有技术）是说明具有交换机软件和交换机硬件的远程通信交换机的框图；

图 3 是示出本发明方法的功能性框图；

图 4 是说明在本发明最佳实施方式中用于建立公用功能性软件单元的操作的流程图；

图 5 是更详细地说明图 4 的建立通用公用软件模板（UCST）、参数定义文档（PDD）和抽象数据类型定义集合（CADTD）的步骤的流程图；

图 6 是更详细地说明图 4 的建立公用功能软件单元 (CS) 的步骤的流程图;

图 7 是说明包括根据本发明的指教从通用公用软件模板 (UCST) 和抽象数据类型定义集合 (CSDTD) 建立的公用功能软件单元的远程通信交换机的框图。

图 3 是示出本发明方法的功能性框图。在领域分析 542 中研究所有类型的远程通信交换机 502。领域分析 542 得出对交换机 502 的 BSH 和 HW 的所有可能组合的公用概念、术语、行为和功能。公用行为和功能是在通用公用软件模板 (UCST) 548、相关联的参数定义文档 (PDD) 544 和相关联的抽象数据类型定义集合 (CADTD) 546 中实现的。PDD544 包括在 UCST548 中所用的参数的章和节。CADTD546 包括多个与 UCST548 的参数相关联的抽象数据类型定义。实现 UCST548 和 CADTD546 的计算机代码具有指示 BSH 依赖性、HW 依赖性或者 BSH 和 HW 两者依赖性的前缀。利用 CADTD546 的子集编译和测试 UCST548。在已经编译和测试 UCST548 之后, 从 UCST548 和 CADTD546 建立 BSH 和 HW 不同组合的多个公用功能软件单元 (CS)。公用功能软件单元 (CS) 是通过从 UCST548 和 CADTD546 复制相关联的计算机代码来建立的。利用标识 CS 将在其中运行的交换机所支持的 BSH 和 HW 的组合的前缀替换存在于所复制的计算机代码中的前缀。例如, 可以通过从 UCST548 和 CADTD546 复制相关联的计算机代码来建立公用功能性软件单元 (CS) 555。利用标识 CS555 将在其中运行的交换机的 BSH 和 HW 的前缀替换存在于所复制的计算机代码中的前缀。

图 4 是大致说明在本发明最佳实施方式中用于建立公用功能性软件单元的操作的流程图。首先, 在步骤 604 标识一组远程通信交换机的承载业务处理程序和硬件类型的所有可能组合。接着, 在步骤 605, 对以上所标识的承载业务处理程序和硬件进行领域分析。在步骤 606 从领域分析建立通用控制软件模板 (UCST)、参数定义文档 (PDD) 和抽象数据类型定义集合 (CADTD)。在步骤 608, 从 UCST 和 CADTD 建立公用功能性软件单元。下面更详细地说明以上各步骤。

在步骤 604，标识一组交换机的承载业务处理程序和不同硬件类型的所有组合。例如，对于一特定组交换机，可能存在承载业务处理程序 1、2、3 和 4 以及硬件类型 A、B、C 和 D。上述承载业务处理程序和硬件类型的可能组合可以如以下表 II 所示存在。

表 II

	BSH1	BSH2	BSH3	BSH4
硬件 A	X	X		
硬件 B			X	X
硬件 C	X			
硬件 D			X	X

因此，承载业务处理程序 1 在类型 A 或 C 的硬件上运行，承载业务处理程序 2 在类型 A 的硬件上运行，承载业务处理程序 3 在类型 B 或 D 的硬件上运行，承载业务处理程序 4 在类型 B 或 D 的硬件上运行。

在步骤 605，进行领域分析，以确定在步骤 604 所确定的组合的公用功能和行为。存在几种可以采用的领域分析技术。一种领域分析技术一般包括（1）召集具有每种承载业务处理程序和硬件类型知识的专家；以及（2）定义公用概念和术语。

图 5 更详细地示出在图 4 步骤 606 建立 UCST、PDD 和 CADTD 的步骤的流程图。首先，在步骤 706 建立描述公用功能和术语的文档。在建立 UCST、PDD 和 CADTD 的过程中，不断更新该文档。在本发明的最佳实施方式中，在步骤 708 使用诸如 Borland C++（Borland 国际公司出品）或 Visual C++（Microsoft 公司出品）这样的面向对象的编程语言来定义和实现对象，每个对象实现公用功能的一个子集。后文中，面向对象的编程语立是指 C++。

在步骤 712，为每个对象建立用于实现公用功能的一个子集的部件功能。一个部件功能是由一个名字和可选变元表表示的独立处理操作。部件功能与 C++ 中为一个类定义的函数类似并与常规软件系统中宣称的

过程和函数相近。

在步骤 716，确定和定义所定义对象的参数。一个参数是一个当唤醒部件功能时由所提供的变量替换的符号。每个参数与具体数据类型定义相关联。从领域分析以及相关联的变元表建立的每个部件功能代表一个公用功能性概念。因此，每个部件功能包括相同的名字和相同个数的变元，而不管所用的 BSH 和硬件类型的组合。

每个变元具有一个由参数定义的相关联的数据类型定义。每个变元的参数也对 BSH 和 HW 的每种组合具有相同的名字。然而，BSH 和 HW 的不同组合需要不同的数据类型定义。因此，参数所定义的数据类型定义对其所需的 BSH 和 HW 组合也必须不同。例如，提供具体 BSH 的 ATM - VC 交换机可能要求一个称为通道号（ChannelNumber）的参数具有（整数，整数）（即，（Integer，Integer））的数据类型定义，分别指示 VPI 和 VCI，而实现相同 BSH 的 ATM - VP 交换机可能要求通道号具有（整数）（即，（Integer））的数据类型定义，指示 VPI。

在步骤 718 建立 PDD。PDD 包括一个参数名字列表，对于 BSH 和 HW 的每种组合具有相关联的数据类型定义。在步骤 722，将前缀加到每个参数上。前缀是为每个所标识的 BSH、HW 类型和 BSH 与 HW 类型的组合而选择的。每个所选择的前缀在保持能够适当标识 BSH、HW 类型或 BSH 与 HW 的组合的同时必须尽可能短。每个所选择的前缀也必须具有相同长度。另外，使用一组三个模板前缀来标识一个参数是否是（1）仅依赖于 BSH；（2）仅依赖于 HW；或者（3）依赖于 BSH 和 HW 两者。也要求模板前缀具有与所选择的前缀相同的长度。再者，每个模板前缀是专用的，并且不可以在部件功能的计算机代码（包括注释）内的任何其他组合中使用。例如，表 I 的模板前缀可以是 BSH、HWR 和 BHW，分别代表仅依赖于 BSH、仅依赖于 HW 和依赖于 BSH 及 HW 的参数。

在步骤 722 将所选择前缀加到 PDD 中的每个参数。然而，在增加所选前缀之前，将每个参数分类为 PDD 内的各节。这些节如下：“类属”、仅仅“BSH 依赖”、仅仅“HW 依赖”和“BSH 及 HW 依赖”。对于所有类型的 BSH 和 HW 都具有相同数据类型定义的参数被分类并放置在

“类属”节中。仅依赖于 BSH 并对所有类型的 BSH 都具有相同数据类型定义的参数被分类并放置在“BSH 依赖”节中。仅依赖于 HW 并对所有类型的 HW 都具有相同数据类型定义的参数被分类和放置在“HW 依赖”节中。依赖于 BSH 和 HW 并对所有类型的 BSH 和 HW 都具有相同数据类型定义的参数被分类和放置在“BSH 和 HW 依赖”节中。另外，对每种类型的 BSH、HW 和 BSH 与 HW 的组合，建立单独的章。

类属参数没有增加到其名字中的前缀。依赖于 BSH 的参数具有增加到其名字的指示 BSH 依赖性的前缀。例如，用于通过交换机建立连接的部件功能可以要求一个称为地址（Address）的参数，表示为地址=[整数，通道号，整数，通道号]（即，Address=[integer, ChannelNumber, integer, ChannelNumber]）。地址参数包括交换机的输入的数字 Id、输入通道号、交换机的输出的数字 Id 和输出通道号。没有前缀增加到代表交换机的输入和输出的数字 Id 的参数上，因为它们是类属的。然而，通道号参数对于每种 BSH 类型具有不同的定义，因此，将前缀“BSH”加到该参数的名字中。另外，地址参数的定义依赖于 BSH 的类型，也具有增加到参数名字的前缀“BSH”。具有增加的前缀的地址类型定义可以表示为“BSH地址[整数，BSH通道号，整数，BSH通道号]”（即，BSHAddress [integer, BSHChannelNumber, integer, BSHChannel Number]）。因此，类型定义 BSH 地址（BSHAddress）是依赖于 BSH 的，对于 BSH 和 HW 的所有组合具有相同的数据类型定义。

依赖于 HW 的参数具有增加到其名字的指示 HW 依赖性的前缀。依赖于 BSH 和 HW 的参数具有增加到其名字的指示 BSH 和 HW 依赖性的前缀。依赖于 BSH 并对每种 BSH 类型具有不同数据类型定义的所有参数被分类并放置在 PDD 中每种类型 BSH 的单独节中。例如，表 I 中的每个 BSH 类型 BSH1 - BSH4 对于依赖于它们的参数在 PDD 中具有单独的节。依赖于 HW 并对每种 HW 类型具有不同数据类型定义的所有参数被分类并放置在 PDD 中每种类型 HW 的单独节中。例如，表 I 中的每个 HW 类型 A - D 对于依赖于它们的参数在 PDD 中具有单独的节。依赖于 BSH 和 HW 并对 BSH 和 HW 的每种组合具有不同类型定义的每个参数被分类并放置在 PDD 中对于每种组合的单独节中。例如，表 I 中标

以“X”的每种 BSH 和 HW 组合对于依赖于它们的参数在 PDD 中具有单独的节。

在步骤 724，定义用于测试 UCST 的模板参数。对于 BSH 和 HW 的不同组合具有不同定义的所有参数被定义为模板参数。模板参数以与以上结合依赖于 BSH 和 HW 的参数所述的方式相同的方式接收一个前缀。依赖于 BSH、HW 及 BSH 和 HW 的模板参数被分类并放置在 PDD 中对于每种依赖性的节中。

在步骤 728 建立 CADTD。在本发明的最佳实施方式中，利用 C++ 建立 CADTD。然而，对于本领域内熟练的技术人员而言，显然可以采用任何面向对象的编程语言。面向对象的编程语言向用户提供建立用户定义的数据类型（后文中称为抽象数据类型（ADT））的能力。ADT 可以包括其他 ADT 和/或已经由编程语言定义的基本数据类型（例如，整数（integer）、浮点（float）、长整数（long integer）和双长型（double））。PDD 中的每个参数由一个 ADT 或基本数据类型定义。

在步骤 750，确定是否利用模拟模板实现 UCST。如果编程语言不支持模板，则通过在步骤 730 向文档中的所有对象增加前缀来实现模拟模板。否则，对文档中的所有对象建立语言模板。诸如 C++ 提供的那样的模板可以用来建立类属部件功能，这些功能具有独立于任何数据类型定义的参数。C++ 编译器利用该类属部件功能（模板）产生实现每个 ADT 唯一组合的计算机代码（非类属部件功能）。换言之，C++ 编译器从事在建立对于 ADT 的每个唯一组合的各部件功能中所涉及的工作。

在步骤 730，将模板前缀增加到文档中所有对象的名字上（即，未曾使用的模板）。这与将前缀增加到参数上的处理类似。将对象分类为或者是类属的、依赖于 BSH 的、依赖于 HW 的、或者是依赖于 BSH 和 HW 的。将一个前缀增加到代表每个对象分类的对象名字上。在步骤 732 定义 UCST 检测实例。对 UCST 中的每个单独对象定义检测实例，以便验证该对象根据其设计发挥作用。

在步骤 736 建立包括 UCST 的计算机代码。根据文档（它包括模板前缀和 CADTD 模板前缀）建立实现 UCST 的所有对象。在本发明最佳实施方式中，用 C++ 将对象实现为类，但是对于本领域内熟练的技术人

员而言，能够采用任何面向对象的计算机编程结构。另外，在步骤 736 建立测试 UCST 所需的计算机代码。在步骤 738，对 UCST 进行测试。UCST 是根据先前定义的每种检测实例得以测试的。

图 6 是更详细说明图 4 中在步骤 608 建立 CS 的步骤的流程图。建立 CS 时，首先在步骤 804 确定 CS 将在其上运行的交换机的 BSH 和 HW 的组合。在步骤 806，从 UCST 复制用于实现 CS 的计算机代码，并且复制相关联的 CADTD 子集。该相关子集包括是类属的、依赖于 BSH 的、依赖于 HW 的所有相关联的 ADT，以及专用于 BSH 和 HW 组合的 ADT。在步骤 808，利用 BSH 和 HW 组合的前缀替换所复制的代码中的模板前缀。在已经替换了所复制的代码中的前缀之后，在步骤 810 保存（存储）所复制的代码。在步骤 812，将所复制的代码编译为能够在远程通信交换机上执行的可执行格式。

对 UCST 的维护直接反映到从其建立的 CS 中。例如，在 UCST 中修正了一个错误或者进行了一次更新，则该更新或修正自动地反映到之后从 UCST 建立的所有 CS 中。

图 7 是说明包括根据本发明的指教从 UCST 和 CADTD 建立的公用功能软件单元 CS1A 408 和 CS2A 410 的远程通信交换机 406 的框图。该远程通信交换机 406 包括公用功能软件单元 CS1A 408 和 CS2A 410，专用软件单元 SPS - 1 - 2 - A，和类型 A 的硬件 414（SH - A）。交换机 406 支持类型 1 和 2 的承载业务处理程序（BSH1 和 BSH2）。CS1A 408 依赖于 BSH1 和类型 A 的硬件。CS2A 410 依赖于 BSH2 和类型 A 的硬件。专用软件单元 SPS - 1 - 2 - A 412 依赖于 BSH1、BSH2 和类型 A 的硬件。

因此认为根据以上所述，本发明的工作和结构是清楚的。尽管所示和所述的方法是较佳的，但是显然易于做出各种变化和修改而不背离如权利要求所限定的本发明的实质和范围。

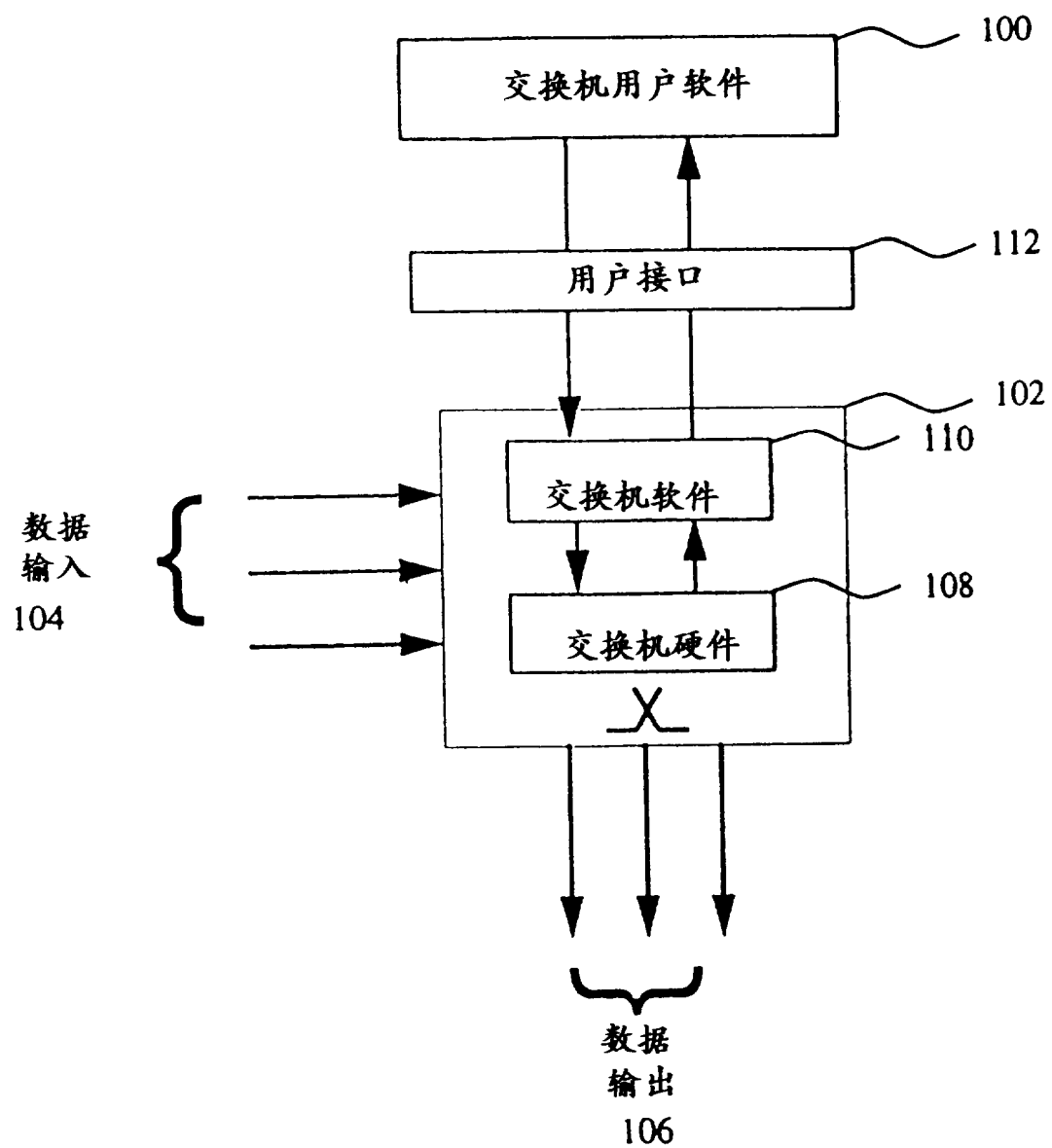


图.1 (现有技术)

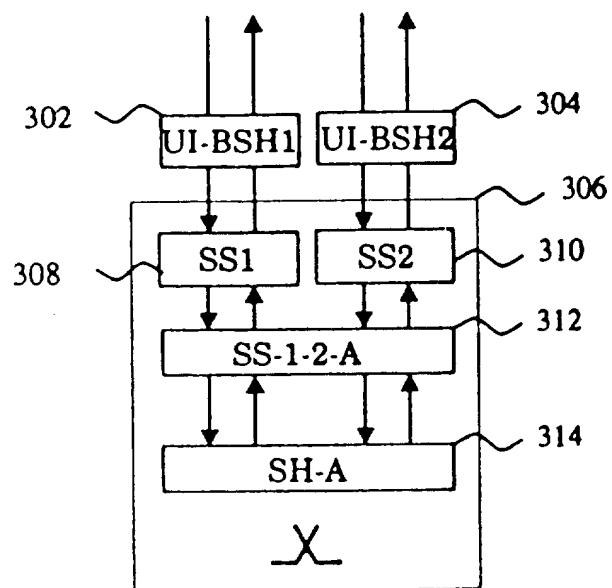


图. 2 现有技术

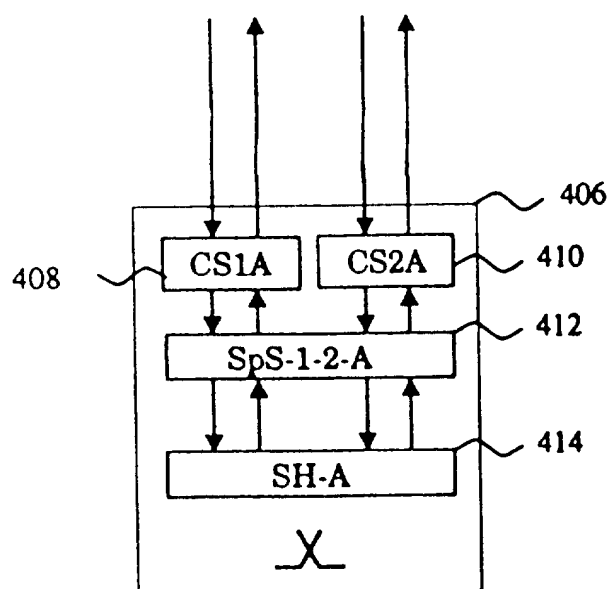


图. 7

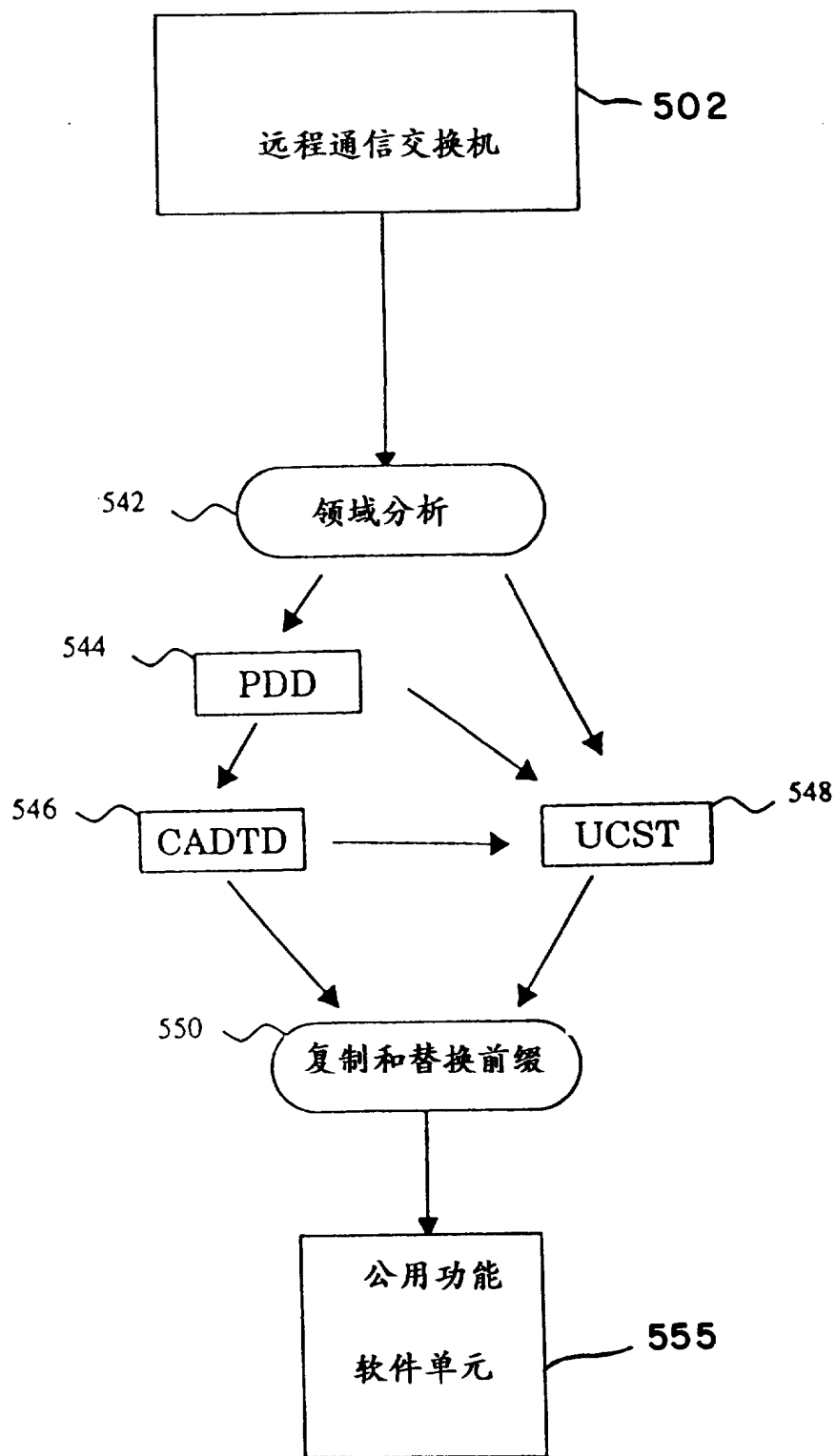


图 3

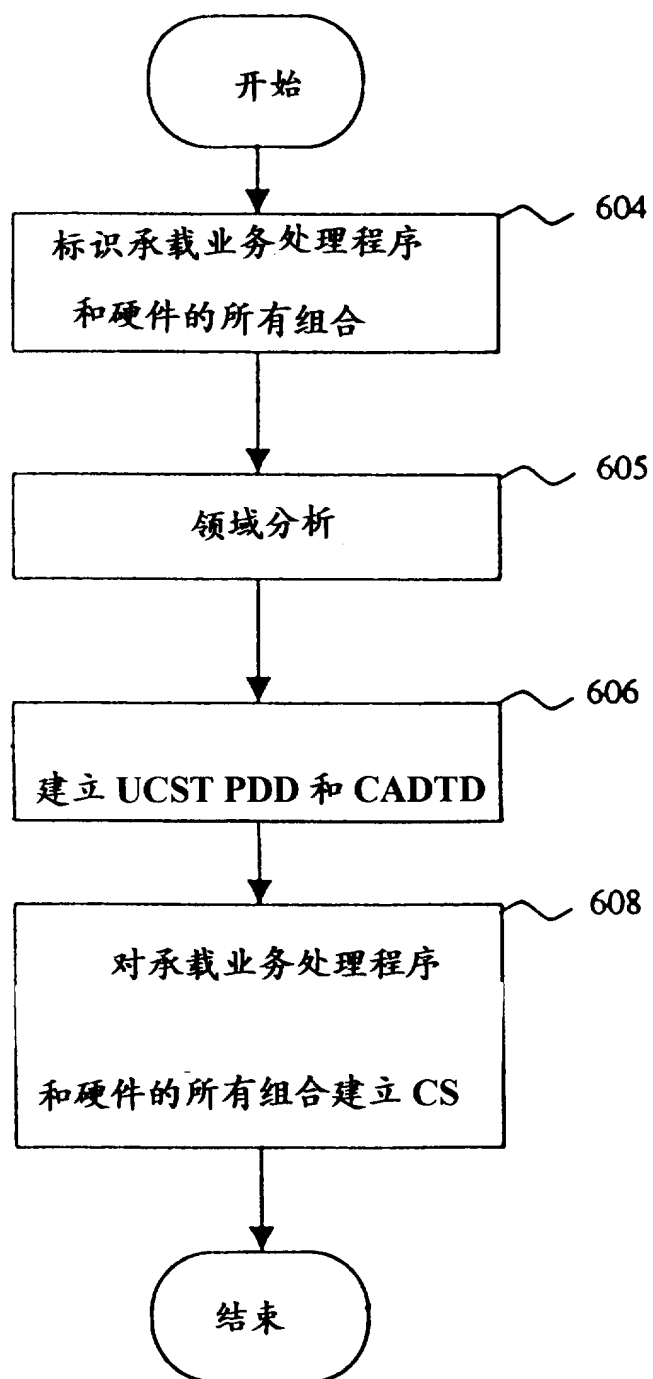


图. 4

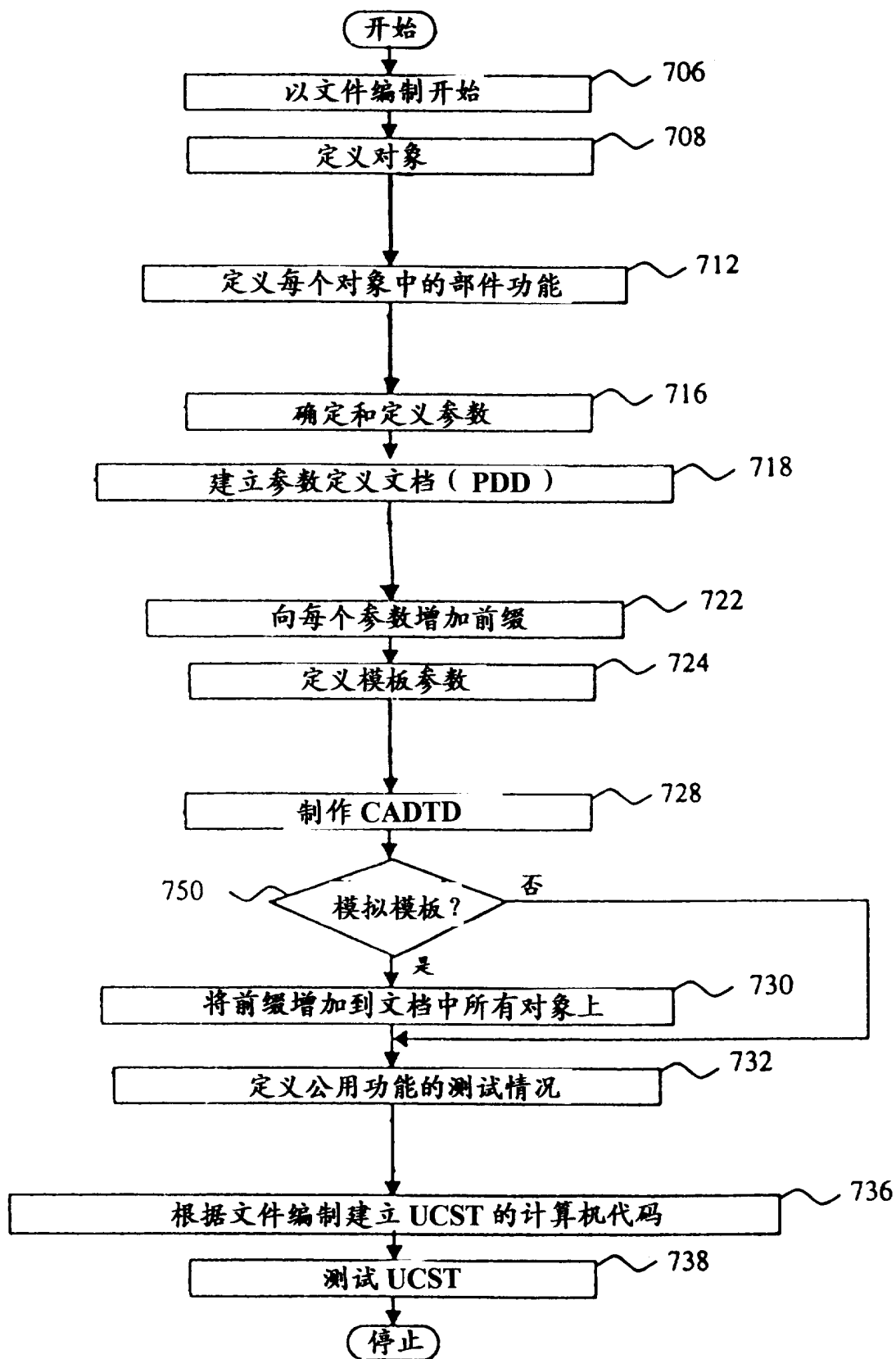


图 5

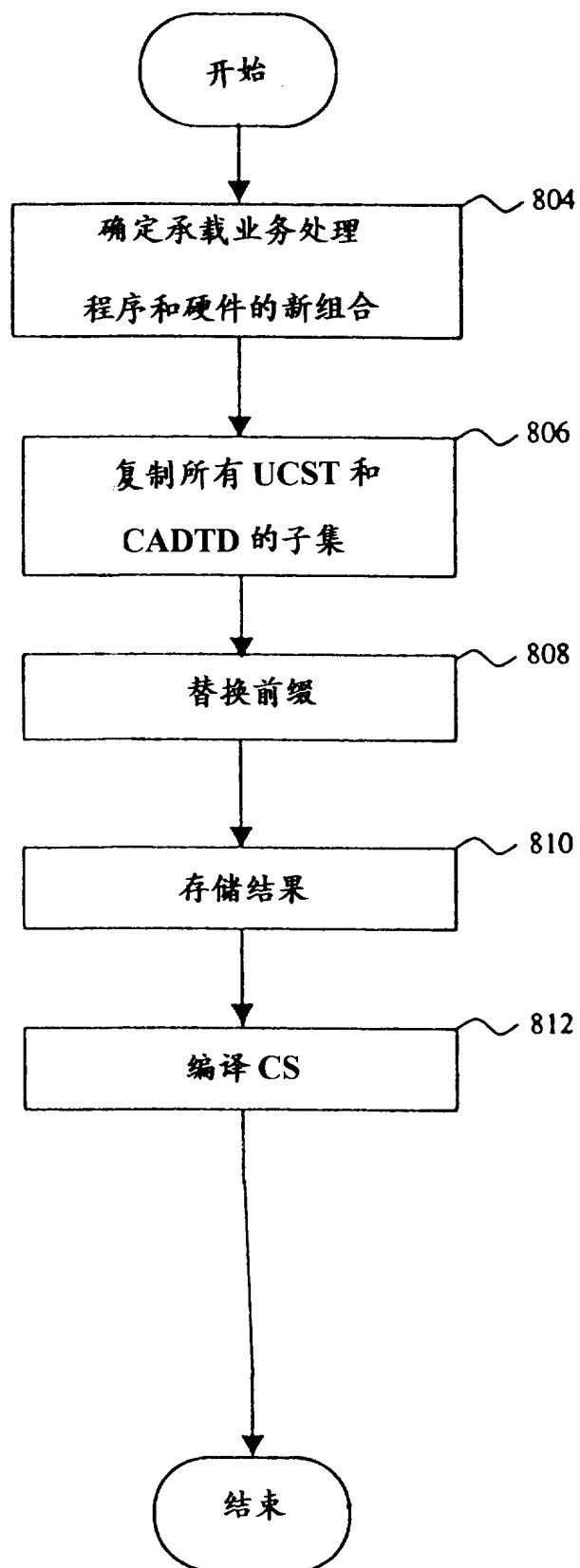


图 6