



# [12] 发明专利说明书

专利号 ZL 200480042860.6

[45] 授权公告日 2009年10月28日

[11] 授权公告号 CN 100555217C

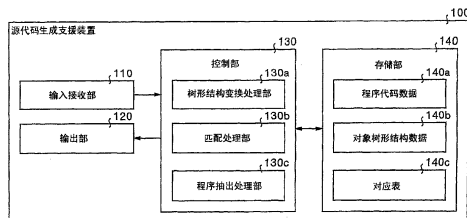
[22] 申请日 2004.4.28  
 [21] 申请号 200480042860.6  
 [86] 国际申请 PCT/JP2004/006168 2004.4.28  
 [87] 国际公布 WO2005/106650 日 2005.11.10  
 [85] 进入国家阶段日期 2006.10.26  
 [73] 专利权人 富士通株式会社  
           地址 日本神奈川县  
 [72] 发明人 宫崎比吕志  
 [56] 参考文献  
       US6070007A 2000.3.30  
       US5671416A 1997.9.23  
       JP7-123452A 1995.5.12  
       US20030212984A1 2003.11.13  
       审查员 韩倩倩

[74] 专利代理机构 隆天国际知识产权代理有限公司  
                     代理人 张龙喆

权利要求书 2 页 说明书 13 页 附图 16 页

[54] 发明名称  
 源代码生成支援装置及源代码生成支援方法

[57] 摘要  
 树形结构变换处理部(130a)基于生成顺序图数据而生成树形结构数据,匹配处理部(130b)从对象树形结构数据(140b)中检索与所生成的树形结构数据相符合的树形结构数据,程序抽出处理部(130c)从相符合的树形结构数据的程序代码确定信息中取得对应的原有的程序代码,将所取得的程序代码的实例名替换为与生成顺序图数据对应的实例名。



1. 一种源代码生成支援装置，对基于面向对象的源代码的生成进行支援，其特征在于，具有：

树形结构生成单元，其基于用于表示要新生成的源代码的规格的顺序图，生成树形结构数据，

检索单元，其在至少存储有原有的源代码和与该原有的源代码对应的树形结构数据的存储装置中，检索与上述树形结构生成单元所生成的树形结构数据相符合的树形结构数据，

取得单元，其从上述存储装置中取得与上述相符合的树形结构数据对应的源代码；

上述树形结构数据是指，抽出包括于上述顺序图中的实例和方法的连接关系，并根据方法的顺序信息和实例的连接信息，由上述树形结构生成单元生成的数据。

2. 根据权利要求 1 所述的源代码生成支援装置，其特征在于，还具有变换处理单元，该变换处理单元将上述取得单元所取得的源代码中所包含的实例变换为与上述顺序图对应的实例。

3. 根据权利要求 1 或 2 所述的源代码生成支援装置，其特征在于，上述树形结构生成单元还具有树形结构生成处理单元，该树形结构生成处理单元基于上述顺序图中所包含的实例和方法的连接关系，生成包含有方法的顺序信息的树形结构数据。

4. 根据权利要求 3 所述的源代码生成支援装置，其特征在于，上述检索单元从上述存储装置中检索包含有与上述树形结构生成单元所生成的树形结构数据中所包含的方法的顺序信息相同的方法的顺序信息的树形结构数据。

5. 一种源代码生成支援方法，对基于面向对象的源代码的生成进行支援，其特征在于，包括：

树形结构生成工序，基于用于表示要新生成的源代码的规格的顺序图，生成树形结构数据，

检索工序，在至少存储有原有的源代码和与该原有的源代码对应的树形结构数据的存储装置中，检索与在上述树形结构生成工序中生成的树形结构

数据相符合的树形结构数据，

取得工序，从上述存储装置中取得与上述相符合的树形结构数据对应的源代码；

上述树形结构数据是指，抽出包括于上述顺序图中的实例和方法的连接关系，并根据方法的顺序信息和实例的连接信息，在上述树形结构生成工序中生成的数据。

6. 根据权利要求 5 所述的源代码生成支援方法，其特征在于，还包括变换处理工序，该变换处理工序将在上述取得工序中取得的源代码中所包含的实例变换为与上述顺序图对应的实例。

7. 根据权利要求 5 或 6 所述的源代码生成支援方法，其特征在于，上述树形结构生成工序还包括树形结构生成处理工序，该树形结构生成处理工序基于上述顺序图中所包含的实例和方法的连接关系，生成包含有方法的顺序信息的树形结构数据。

8. 根据权利要求 7 所述的源代码生成支援方法，其特征在于，上述检索工序从上述存储装置中检索包含有与在上述树形结构生成工序中生成的树形结构数据中所包含的方法的顺序信息相同的方法的顺序信息的树形结构数据。

## 源代码生成支援装置及源代码生成支援方法

### 技术领域

本发明涉及一种对基于面向对象的源代码的生成进行支援的源代码生成支援装置及源代码生成支援方法，尤其是涉及一种能够对原有的源代码进行再利用而提高程序开发效率的源代码生成支援装置及源代码生成支援方法。

### 背景技术

最近，为了提高程序开发效率，正在基于面向对象而生成程序代码。在该面向对象中，集成数据和处理作为一个对象，以向该对象发送激活处理的消息的方式，编写程序代码。

另一方面，正在尝试根据用户所希望的程序代码的规格书等，自动生成与规格书对应的程序代码，但因为生成程序代码的处理逻辑不完备，所以只能生成用户所希望的程序代码的雏形而已（例如，参照专利文献1）。即，为了生成完备的程序代码，用户必须利用简易的脚本语言等，重新添写程序代码。

为了解决这个问题，正在考虑这样的技术：将程序代码和该程序代码的结构图建立关联而预先生成，当用户编辑结构图时，对该结构图进行的编辑就会被反映在对应的程序代码上，从而能够生成新的程序代码（例如，参照专利文献2）。另外，在专利文献3中，公开了自动管理有关程序代码的规格书的匹配性的技术。

专利文献1（JP特开平11-219287号公报）

专利文献2（JP专利3299259号公报）

专利文献3（JP特开平11-3216号公报）

然而，在相关的现有的技术中，需要预先记入跟想要新生成的程序代码同等的程序代码，用户的负担很大。

另外，在生成与预先生成的程序代码的结构图不对应的新的程序代码的情况下，就不能编辑结构图而生成新的程序代码，用户自己需要生成新的程

序代码，生成程序代码的效率不高。

本发明是为了解决上述现有技术的问题，其目的在于提供一种能够解除用户的负担、能够高效率地生成程序代码的源代码生成支援装置及源代码生成支援方法。

## 发明内容

为了解决上述问题而达到目的，本发明是一种源代码生成支援程序，对基于面向对象的源代码的生成进行支援，其特征在于，使计算机执行以下步骤：树形结构生成步骤，基于表示要新生成的源代码的规格的顺序图，生成树形结构数据；检索步骤，从至少存储有原有的源代码、和与该原有的源代码对应的树形结构数据的存储装置中，检索与在上述树形结构生成步骤中生成的树形结构数据相符合的树形结构数据；取得步骤，从上述存储装置中取得与上述相符合的树形结构数据对应的源代码。

另外，本发明是一种源代码生成支援方法，对基于面向对象的源代码的生成进行支援，其特征在于，包括：树形结构生成工序，基于表示要新生成的源代码的规格的顺序图，生成树形结构数据；检索工序，从至少存储有原有的源代码、和与该原有的源代码对应的树形结构数据的存储装置中，检索与在上述树形结构生成工序中生成的树形结构数据相符合的树形结构数据；取得工序，从上述存储装置中取得与上述相符合的树形结构数据对应的源代码。

本发明提供一种源代码生成支援装置，对基于面向对象的源代码的生成进行支援，其特征在于，具有：树形结构生成单元，其基于用于表示要新生成的源代码的规格的顺序图，生成树形结构数据；检索单元，其在至少存储有原有的源代码和与该原有的源代码对应的树形结构数据的存储装置中，检索与上述树形结构生成单元所生成的树形结构数据相符合的树形结构数据；取得单元，其从上述存储装置中取得与上述相符合的树形结构数据对应的源代码。上述树形结构数据是指，抽出包括于上述顺序图中的实例和方法的连接关系，并根据方法的顺序信息和实例的连接信息，由上述树形结构生成单元生成的数据。

本发明提供一种源代码生成支援方法，对基于面向对象的源代码的生成

进行支援，其特征在于，包括：树形结构生成工序，基于用于表示要新生成的源代码的规格的顺序图，生成树形结构数据；检索工序，在至少存储有原有的源代码和与该原有的源代码对应的树形结构数据的存储装置中，检索与在上述树形结构生成工序中生成的树形结构数据相符合的树形结构数据；取得工序，从上述存储装置中取得与上述相符合的树形结构数据对应的源代码。上述树形结构数据是指，抽出包括于上述顺序图中的实例和方法的连接关系，并根据方法的顺序信息和实例的连接信息，在上述树形结构生成工序中生成的数据。

根据该发明，基于表示要新生成的源代码的规格的顺序图而生成树形结构数据，从至少存储有原有的源代码和与该原有的源代码对应的树形结构数据的存储装置中检索与生成的树形结构数据相符合的树形结构数据，从存储装置中取得与相符合的树形结构数据对应的源代码，所以不需要完全从零开始生成源代码，而只要生成顺序图，就能够自动生成所希望的源代码，从而能够消除用户的负担，提高程序开发效率。

## 附图说明

图 1 是表示本实施方式中的源代码生成装置进行的程序再利用的概念的图，图 2 是表示源代码生成支援装置的结构的功能方框图，图 3 是表示顺序图的列表结构的一个例子的图，图 4 是表示树形结构数据的数据结构的一个例子的图，图 5 是表示以在图 3 所示的顺序图数据为基础而生成的树形结构数据的图，图 6 是表示对应表的一个例子的图，图 7 是说明对各树形结构数据的单元的识别号码的分配的图，图 8 是表示生成顺序图和生成顺序图的树形结构数据的图，图 9 是对程序代码的实例名的变换进行说明的图，图 10 是表示准备处理的处理步骤的流程图，图 11 是表示树形结构变换处理的处理步骤的流程图，图 12 是表示程序生成处理的处理步骤的流程图，图 13 是表示匹配处理的处理步骤的流程图，图 14 是表示单元选择处理的处理步骤的流程图，图 15 是表示执行本实施方式涉及的源代码生成支援程序的计算机系统的图，图 16 是表示图 15 所示的主体部 501 的结构的功能方框图。

## 具体实施方式

---

以下，基于附图详细地说明本发明的源代码生成支援程序及源代码生成支援方法的实施方式。另外，本发明并不限制于该实施方式。另外，在本实

施方式中，虽然对于物理上独立地构成各个功能的源代码生成支援装置进行说明，但作为源代码生成支援程序，也可以通过以用软件实现各个功能来实施。

## 实施方式

首先，针对本实施方式涉及的源代码生成支援装置进行的程序再利用的概念进行说明。图 1 是表示本实施方式涉及的源代码生成装置进行的程序再利用的概念的图。另外，在本实施方式中，利用面向对象而生成程序代码。

本实施方式所示的源代码生成支援装置预先取得原有的程序代码和表示该程序代码的规格的顺序图，根据所取得的顺序图生成树形结构数据，将树形结构数据和程序代码建立关联而预先存储。

在这里，所谓树形结构数据，就是将包含于顺序图中的实例和方法的连接关系抽出、而根据方法的顺序信息和实例的连接信息生成的数据。另外，实例就是在执行程序的情况下以类的形式而生成的，方法表示在各个实例之间进行的过程。

然后，在再利用原有的程序代码而生成新的程序代码的情况下，取得要新生成的程序代码的顺序图，根据所取得的顺序图生成树形结构数据，从预先生成的树形结构数据中检索与所生成的树形结构数据相符合的树形结构数据。

然后，将跟相符合的树形结构数据对应的程序代码抽出，将包含于所抽出的程序代码中的实例名替换为新的实例名，从而能够生成新的程序代码。

接着，针对源代码生成支援装置的结构进行说明。图 2 是表示源代码生成支援装置 100 的结构的功能方框图。

如该图所示，该源代码生成支援装置 100 具备输入接收部 110、输出部 120、控制部 130 和存储部 140。

输入接收部 110 从未图示的键盘等的输入装置取得顺序图的数据、程序代码等，将所取得的顺序图的数据和程序代码提交给控制部 130。另外，输入接收部 110 将顺序图的数据变换成规定的数据结构（以下，标记为顺序图数据），提交给控制部 130。

图 3 是表示顺序图数据的列表结构的一个例子的图。如该图所示，顺序



图数据包括实例表 210 和方法箭头表 220。另外，200 表示以顺序图数据表示的顺序图。

实例表 210 包括“左上坐标”、“宽度”、“高度”、“实例信息”、“生命线长度”和“生命线坐标”。

在这里，“左上坐标”示出表示实例的方框 200a 的左上角的坐标，“宽度”示出表示实例的方框 200a 的宽度，“高度”示出表示实例的方框 200a 的高度。

另外，“实例信息”示出存储在表示实例的方框 200a 中的实例名和包含有该实例名的类名，“生命线长度”表示生命线 200b 的长度，“生命线坐标”表示生命线 200b 的顶点的坐标。

例如，存在于左端的实例的方框以及表示生命线的实例表的“左上坐标”是“(x1,y1)”，“宽度”是“w1”，“高度”是“h1”，“实例信息”是“a:Order”，“生命线长度”是“L1”，“生命线坐标”是“Lx1,Ly1”。另外，省略其它的实例表的说明。

接着，针对方法箭头表 220 进行说明。如该图所示，方法箭头表 220 包括“起始点坐标”、“方法名”、“长度”、“线型”、“连接起点类”和“连接目标类”。

在这里，“起始点坐标”表示箭头 200c 的左端的坐标，“方法名”表示以箭头表示的方法的名字，“长度”表示箭头 200c 的长度，“线型”表示箭头 200c 的线的种类，“连接起点类”表示箭头 200c 的连接起点的类名，“连接目标类”表示箭头 200c 的连接目标的类名。

例如，存在于左端的箭头的方法箭头表的“起始点坐标”是“(a1,b1)”，“方法名”是“recieveOrder”，“长度”是“l1”，“线型”是“Solid”，“连接起点类”是“无”，“连接目标类”是“a:Order”。另外，省略其它的方法箭头表的说明。

输出部 120 从控制部 130 接收程序代码，并输出所接收的程序代码。

控制部 130 是生成树形结构数据、再利用原有的程序代码而生成新的程序代码的处理部，具有树形结构变换处理部 130a、匹配处理部 130b、和程序抽出处理部 130c。

树形结构变换处理部 130a 从输入接收部 110 接收顺序图数据和程序代码，并将接收到的顺序图数据变换成树形结构数据。然后，树形结构变换处理部 130a 将程序代码和顺序图数据分别作为程序代码数据 140a 以及对象树形结构数据 140b 而存储在存储部 140 中。

另外，树形结构变换处理部 130a 在将程序代码和树形结构数据存储存储在存储部 140 中的情况下，对树形结构数据添加用于确定对应的程序代码存储在何处的确定信息（以下，标记为程序代码确定信息）。

图 4 是表示树形结构数据的数据结构的一个例子的图。该图所示的树形结构数据具有程序代码确定信息区域、方法名区域、连接目标类名区域和指针区域。以下，将方法名区域、连接目标类名区域和指针区域合在一起标记为单元（另外，在树形结构数据的前头的单元中包含有程序代码确定信息区域）。

在这里，程序代码确定信息区域是存储程序代码确定信息的区域，方法名区域是存储方法名的区域，连接目标类名区域是存储连接目标的类名的区域，指针区域存储成为下一个连接目标的单元的地址。另外，在下一个连接目标存在多个的情况下，在指针区域存储多个地址。

另外，当根据图 3 所示的顺序图数据生成树形结构数据时，可如图 5 的 300 所示；当图示树形结构数据 300 时，可如 310 所示。

另外，树形结构数据 300 包括单元 300a~300d。在单元 300a 的程序代码确定信息区域存储“程序代码确定信息”，在方法名区域存储“RecieveOrder”，在连接目标类名区域存储“Order”，在指针区域存储“单元 300b 所处的地址”。

在单元 300b 的方法名区域存储“createOrder”，在连接目标类名区域存储“Receipt”，在指针区域存储“单元 300c 以及 300d 所处的地址”。

在单元 300c 的方法名区域存储“getCustomer”，在连接目标类名区域存储“Customer”。另外，因为不存在后续的单元，所以在单元 300c 的指针区域不存储数据。

在单元 300d 的方法名区域存储“getProduct”，在连接目标类名区域存储“Product”。因为不存在后续的单元，所以与单元 300c 同样地在单元 300d 的指针区域不存储数据。

另外，树形结构变换处理部 130a 基于包含于顺序图数据的实例中的实例名和类名，作成表示各个类名和实例名的对应关系的对应表，将所作成的对应表 140c 存储在存储部 140 中。

图 6 是表示对应表的一个例子的图。如该图所示，对应表 140c 由类名

和实例名构成。

具体地说，表示为，与类名“Order”对应的实例名是“a”，与类名“Receipt”对应的实例名是“b”，与类名“Product”对应的实例名是“c”，与类名“Customer”对应的实例名是“d”。该对应表将被后述的程序抽出处理部 130c 所利用。

然而，树形结构变换处理部 130a 再利用原有的程序代码，在接收到表示要新生成的程序代码的规格的顺序图数据（以下，标记为生成顺序图数据）的情况下，将接收到的生成顺序图数据变换成树形结构数据，并将所变换的树形结构数据提交给匹配处理部 130b，然后将生成顺序图数据提交给程序抽出处理部 130c。

匹配处理部 130b 在从树形结构变换处理部 130a 接收到树形结构数据的情况下，从存储部 140 的对象树形结构数据 140b 中检索与接收到的树形结构数据相符合的树形结构数据 140b。

具体地说，如图 7 所示，对从树形结构变换处理部 130a 接收到的树形结构数据以及从对象树形结构数据 140b 取得的树形结构数据中所包含的单元的连接关系进行分析，从而对各树形结构数据的各单元分配识别号码。

识别号码的分配步骤如下：将识别号码“1”分配给树形结构数据的前头的单元，将识别号码“2”分配给分配了识别号码“1”的下一个单元。

另外，在多个单元连接在同一层的情况下，按照连接起点的单元的指针中所包含的地址被存储的顺序分配识别号码。

例如，对于图 5 所示的树形结构数据 300，按照单元 300a、单元 300b、单元 300c 和单元 300d 的顺序分配识别号码 1、2、3、4。

然后，对于识别号码相同的单元，按顺序判断方法名以及连接目标类名是否相同，当所有的单元的方法名以及连接目标类名相同时，判断为树形结构数据相符合。然后，将相符合的树形结构数据提交给程序抽出处理部 130c。

例如，因为与图 8 所示的生成顺序图 350 对应的树形结构数据 360 和图 5 的树形结构数据 300 相符合，所以匹配处理部 130b 将树形结构数据 300 提交给程序抽出处理部 130c。

程序抽出处理部 130c 从匹配处理部 130b 取得树形结构数据，基于包含于该取得的树形结构数据中的程序代码指定信息，从程序代码数据 140a 中检索与树形结构数据对应的程序代码。

然后，程序抽出处理部 130c 基于检索出的程序代码和对应表 140c，将包含于程序代码中的实例名替换为新的实例名，并将该替换的程序代码提交给输出部 120。

例如，程序抽出处理部 130c 接收到树形结构数据 300 时，能够从包含于树形结构数据 300 中的程序代码确定信息中抽出与树形结构数据 300 对应的程序代码 400b（如图 9 所示）。

然后，当程序抽出处理部 130c 基于对应表 140c 而检索与程序代码 400b 的实例“b”对应的类时，判明与实例“b”对应的类是“Receipt”。

因此，因为在图 8 所示的生成顺序图数据的类“Receipt”的实例是“y”，所以程序抽出处理部 130c 将程序代码 400a 的实例从“b”变换为“y”，从而生成新的程序代码 400b。

即，不需要从零开始生成程序代码，只要生成要新生成的程序代码的规格的顺序图，就能够生成与顺序图对应的程序代码。

另外，虽然在本实施方式中省略详细说明，但在对从树形结构变换处理部 130a 接受的树形结构数据分配的认识号码的数、和对从对象树形结构数据 140b 取得的树形结构数据分配的认识号码的数不一致时，判断为树形结构数据不相符。

接着，针对树形结构变换处理部 130a 预先在存储部 140 中存储树形结构数据和与树形结构数据对应的程序代码的准备处理进行说明。图 10 是表示准备处理的处理步骤的流程图。

如图 10 所示，输入接收部 110 读入程序代码以及顺序图（步骤 S101），将程序代码以及顺序图数据提交给树形结构变换处理部 130a（步骤 S102）。

然后，树形结构变换处理部 130a 进行树形结构变换处理（步骤 S103），作成对应表（步骤 S104），将对应表存储到存储部 140 中（步骤 S105）。

然后，树形结构变换处理部 130a 将程序代码存储在存储部 140 中（步骤 S106），对树形结构数据追加程序代码确定信息（步骤 S107），将树形结构数据存储在存储部 140 中（步骤 S108）。

接着，针对图 10 的步骤 S103 所示的树形结构变换处理进行说明。图 11 是表示树形结构变换处理的处理步骤的流程图。如该图所示，树形结构变换处理部 130a 检索顺序图的最左边且最上边的方法箭头（步骤 S201），生成

单元(步骤 S202),将方法名以及连接目标的类名存储在单元中(步骤 S203),并存储箭头的连接起点类名(步骤 S204)。(虽然在上述的说明中从略,但树形结构变换处理部 130a 也存储所检索到的箭头的连接起点类名)

然后,检索下方向最近的下一个箭头(步骤 S205),判断箭头是否存在(步骤 S206),在箭头不存在时(在步骤 S206 为否),结束树形结构变换处理。

另一方面,箭头存在时(在步骤 S206 为是),生成单元(步骤 S207),将方法名以及连接目标的类名存储在单元(步骤 S208),判断与前一个箭头相关的连接起点类名和与当前的箭头相关的连接起点类名是否相同(步骤 S209)。

当与前一个箭头相关的连接起点类名和与当前的箭头相关的连接起点类名不不同时(在步骤 S209 为否),将当前的单元的地址存储在前一个生成的单元的指针中(步骤 S210),将所存储的箭头的连接起点类名更新为当前的箭头的连接起点类名(步骤 S211),转移到步骤 S205。

另一方面,当与前一个箭头相关的连接起点类名和与当前的箭头相关的连接起点类名相同时(在步骤 S209 中为是),将当前的单元的地址追加在调出起点的单元的指针中(步骤 S212),转移到步骤 S211。

接着,针对基于要新生成的程序代码的顺序图而生成程序代码的程序生成处理进行说明。图 12 是表示程序生成处理的处理步骤的流程图。如该图所示,输入接收部 110 读入顺序图(步骤 S301),将顺序图数据提交给树形结构变换处理部 130a(步骤 S302),树形结构变换处理部 130a 进行树形结构变换处理(步骤 S303),将树形结构数据提交给匹配处理部 130b(步骤 S304)。

然后,匹配处理部 130b 进行匹配处理(步骤 S305),判定相符合的树形结构数据是否存在(步骤 S306),当相符合的树形结构数据不存在时(在步骤 S306 中为否),结束处理。

另一方面,当相符合的树形结构数据存在时(在步骤 S306 中为是),匹配处理部 130b 将相符合的树形结构数据提交给程序抽出处理部 130c(步骤 S307),程序抽出处理部 130c 检索与相符合的树形结构数据对应的程序代码(步骤 S308),根据对应表来替换程序代码的实例名(步骤 S309),

将程序代码提交给输出部 120（步骤 S310），输出部 120 输出程序代码（步骤 S311）。

在这里，图 12 的步骤 S303 所示的树形结构变换处理与图 11 所示的树形结构变换处理相同，所以省略说明。

接着，针对图 12 的步骤 S305 所示的匹配处理进行说明。图 13 是表示匹配处理的处理步骤的流程图。如该图所示，匹配处理部 130b 从对象树形结构数据 140b 中取得树形结构数据（步骤 S401），将识别号码分配给树形结构数据（步骤 S402），进行单元选择处理（步骤 S403），判定各个方法名以及连接目标类名是否相同（步骤 S404）。

当各个方法名以及连接目标类名相同时（在步骤 S404 中为是），判定是否选择了包含于树形结构数据中的所有的单元（步骤 S405），如果没选择包含于树形结构数据中的所有的单元（在步骤 S405 中为否），转移到步骤 S403，当选择了包含于树形结构数据中的所有的单元时（在步骤 S405 中为是），判断为一致的树形结构数据存在（步骤 S406）。

另一方面，当各个方法名以及连接目标类名不不同时（在步骤 S404 中为否），判定是否对于所有的树形结构数据进行了匹配（步骤 S407），当没有对所有的树形结构数据进行匹配时（在步骤 S407 中为否），从对象树形结构数据取得下一个树形结构数据（步骤 S408），转移到步骤 S402。

另一方面，当对所有的树形结构数据进行了匹配时（在步骤 S407 中为是），判断为一致的树形结构数据不存在（步骤 S409）。

接着，针对图 13 的步骤 S403 所示的单元选择处理进行说明。图 14 是表示单元选择处理的处理步骤的流程图。如该图所示，匹配处理部 130b 判定是否第一次选择从树形结构变换处理部 130a 接收的树形结构数据以及从对象树形结构数据 140b 取得的树形结构数据中所包含的单元（步骤 S501），当是第一次选择单元时（在步骤 S501 中为是），分别选择与识别号码“1”对应的单元（步骤 S502）。

另一方面，当不是第一次选择单元时（在步骤 S501 中为否），判定是否从对象树形结构数据 140b 新取得到树形结构数据（步骤 S503），当从对象树形结构数据新取得到树形结构数据时（在步骤 S503 中为是），将以前从对象树形结构数据 140b 取得到的旧树形结构数据变换为新取得到的树形

结构数据（步骤 S504），转移到步骤 S502。

另一方面，当没有从对象树形结构数据新得到树形结构数据时（在步骤 S503 中为否），分别选择与上次选择的识别号码的下一个识别号码对应的单元（步骤 S505）。

如上所述，本实施方式涉及的源代码生成支援程序，因为树形结构变换处理部 130a 根据生成顺序图数据而生成树形结构数据，匹配处理部 130b 从对象树形结构数据 140b 中检索与生成的树形结构数据相符合的树形结构数据，程序抽出处理部 130c 从相符合的树形结构数据的程序代码确定信息中取得所对应的原有的程序代码，将取得的程序代码的实例名替换为与生成顺序图数据对应的实例名，从而不需要完全从零开始生成程序代码，而只要生成顺序图，就能够生成程序代码，进而能够提高程序代码的生产性。

另外，因为只要生成顺序图，就能够自动生成与顺序图对应的程序代码，所以即使用户没有关于脚本语言的语法的知识，也能够容易的生成程序代码。

另外，虽然在本实施方式中针对源代码生成支援装置 100 进行了说明，但也可以通过软件来实现具有该源代码生成支援装置 100 的结构，从而能够得到具有同样功能的源代码生成支援程序。因此，针对执行该源代码生成支援程序的计算机系统进行了说明。

图 15 是表示执行本实施方式涉及的源代码生成支援程序的计算机系统的图。如该图所示，该计算机系统 500 具有主体部 501、根据来自主体部 501 的指示而在显示画面 502a 上显示信息的显示器 502、用于对该计算机系统 500 输入各种的信息的键盘 503、用于对在显示器 502 的显示画面 502a 上的任意位置进行指定的鼠标 504、用于连接到 LAN506 或者广域网（WAN）的 LAN 接口、和用于连接到公用线 507 的调制解调器 505。在这里，LAN506 将其他计算机系统（PC）511、服务器 512、打印机 513 等和计算机系统 500 连接在一起。

另外，图 16 是表示图 15 所示的主体部 501 的结构的功能方框图。如该图所示，该主体部 501 具有 CPU521、RAM522、ROM523、硬盘驱动器（HDD）524、CD-ROM 驱动器 525、FD 驱动器 526、I/O 接口 527 和 LAN 接口 528。

并且，在该计算机系统 500 中被执行的源代码生成支援程序存储在软盘

(FD) 508、CD-ROM509、DVD 盘、光磁盘、IC 卡等的便携式存储介质中，被从这些存储介质中读出而安装在计算机系统 500 上。然后，所安装的视频发送程序存储在 HDD524 中，可利用 RAM522、ROM523 等，由 CPU521 执行。

如上所说明的那样，根据本发明，基于表示要新生成的源代码的规格的顺序图而生成树形结构数据，从至少存储有原有的源代码和与该原有的源代码对应的树形结构数据的存储装置中检索与所生成的树形结构数据相符合的树形结构数据，从存储装置中取得与相符合的树形结构数据对应的源代码，所以不需要完全从零开始生成源代码，而只要生成顺序图，就能够自动生成所希望的源代码，从而能够消除用户的负担，提高程序开发效率。

#### 工业上的可利用性

如上所述，本发明的源代码生成支援程序以及源代码生成支援方法适合需要利用原有的程序代码来高效率的生成源代码的基于面向对象的程序开发系统等中。



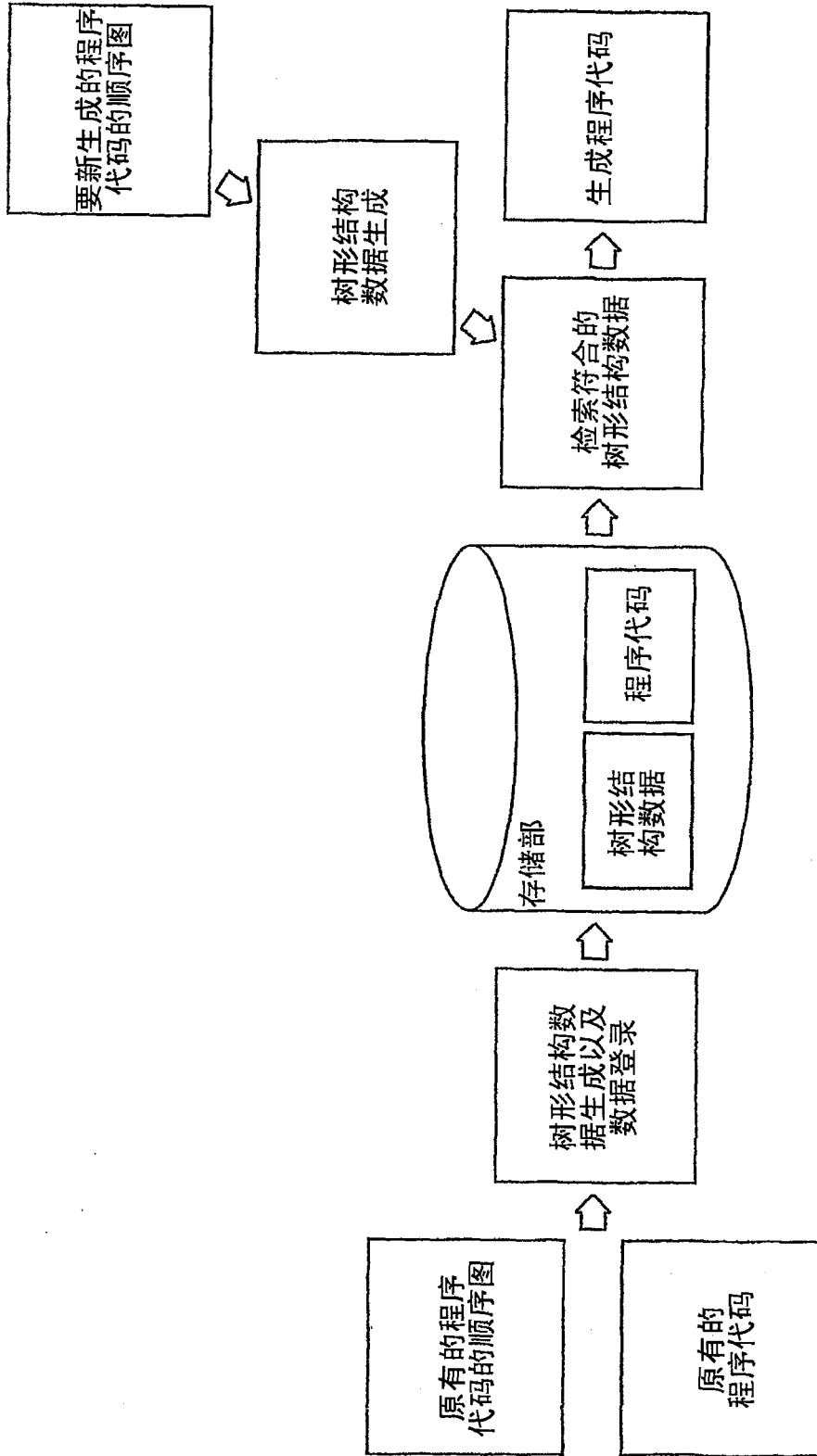


图1

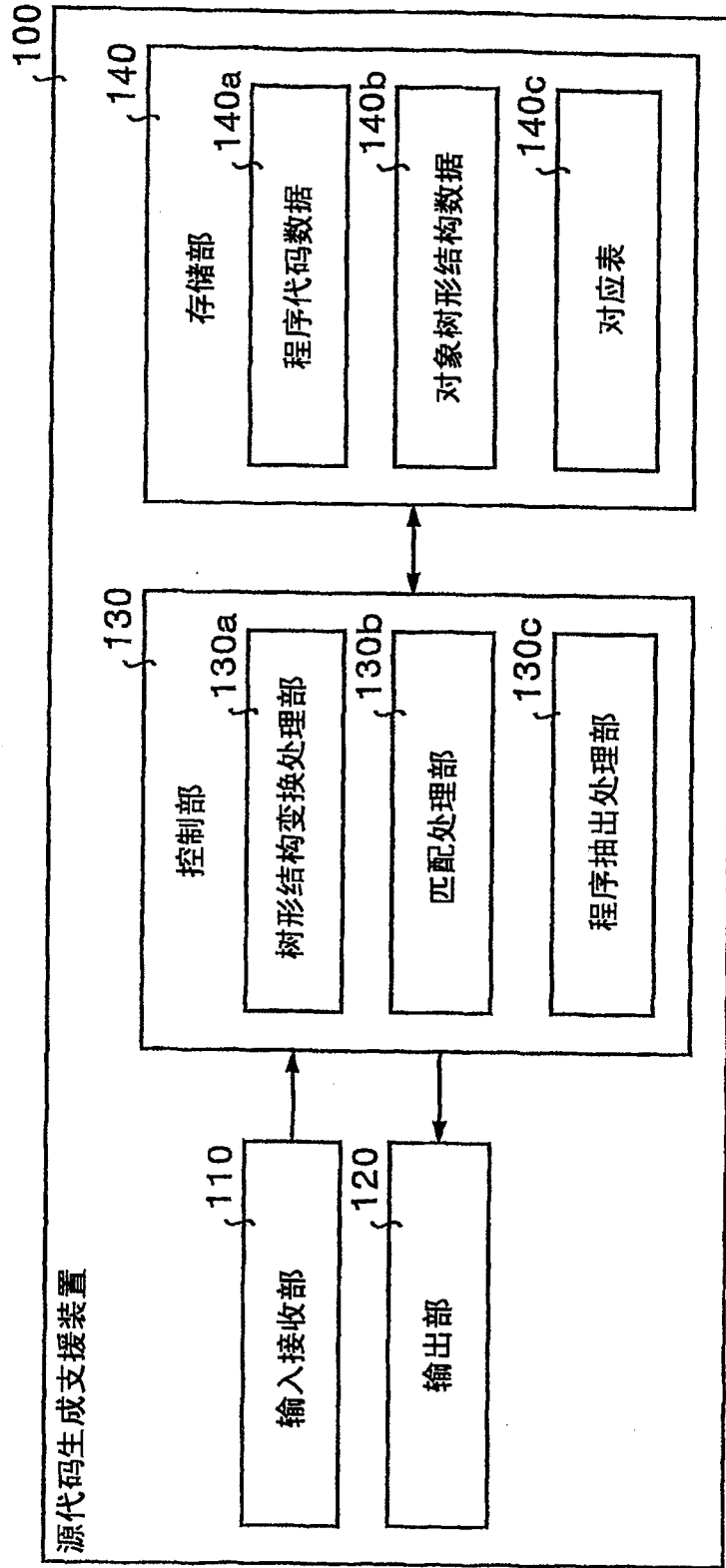
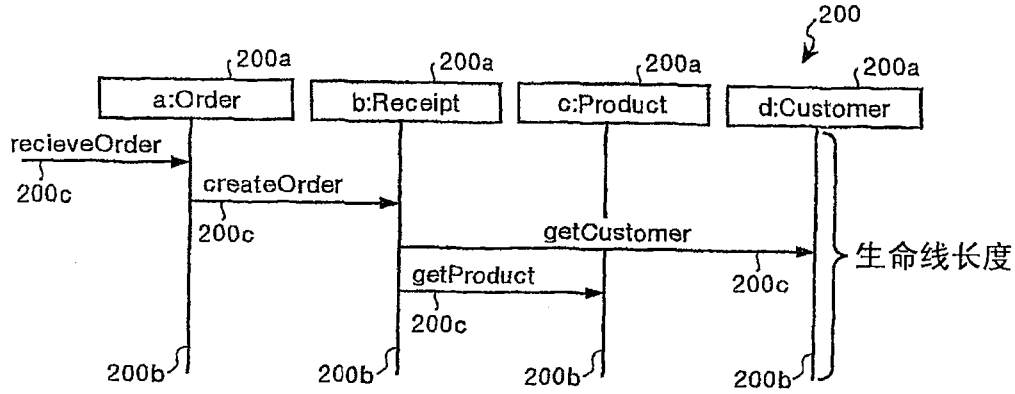


图 2



左上坐标	(x1,y1)	(x2,y2)	(x3,y3)	(x4,y4)
宽度	w1	w2	w3	w4
高度	h1	h2	h3	h4
实例信息	a:Order	b:Receipt	c:Product	d:Custom
生命线长度	L1	L2	L3	L4
生命线坐标	Lx1,Ly1	Lx2,Ly2	Lx3,Ly3	Lx4,Ly4

[ 实例列表 ]

起始点坐标	(a1,b1)	(a2,b2)	(a3,b3)	(a4,b4)
方法名	recieveOrder	createOrder	getCustomer	getProduct
长度	I1	I2	I3	I4
线型	Solid	Solid	Solid	Solid
连接起点类	-	a:Order	b:Receipt	b:Receipt
连接目标类	a:Order	b:Receipt	d:Customor	c:Product

[ 方法箭头列表 ]

图 3

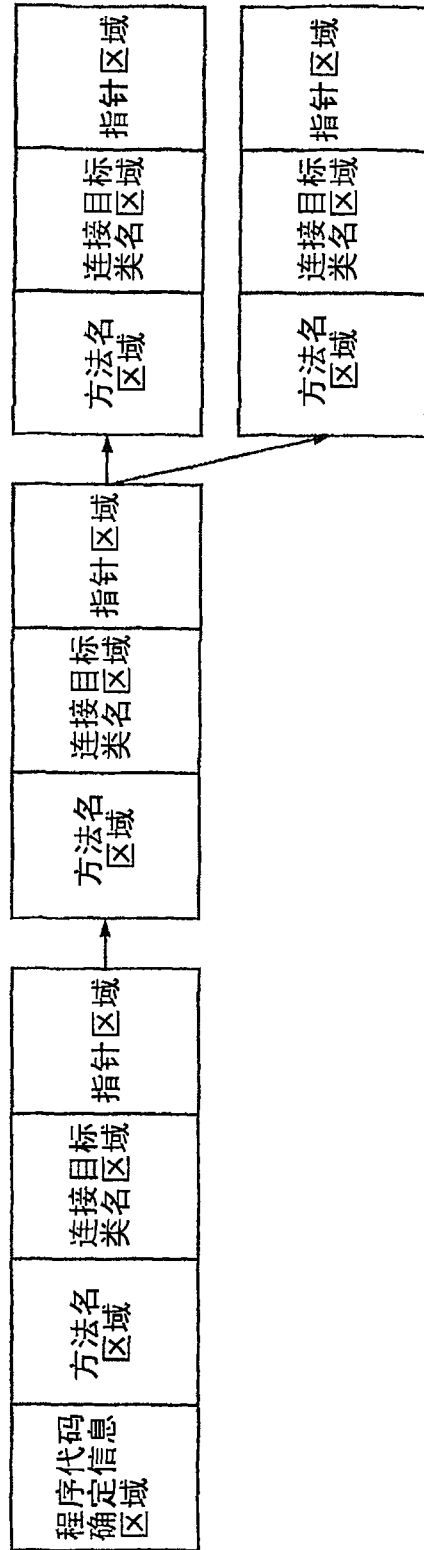


图 4

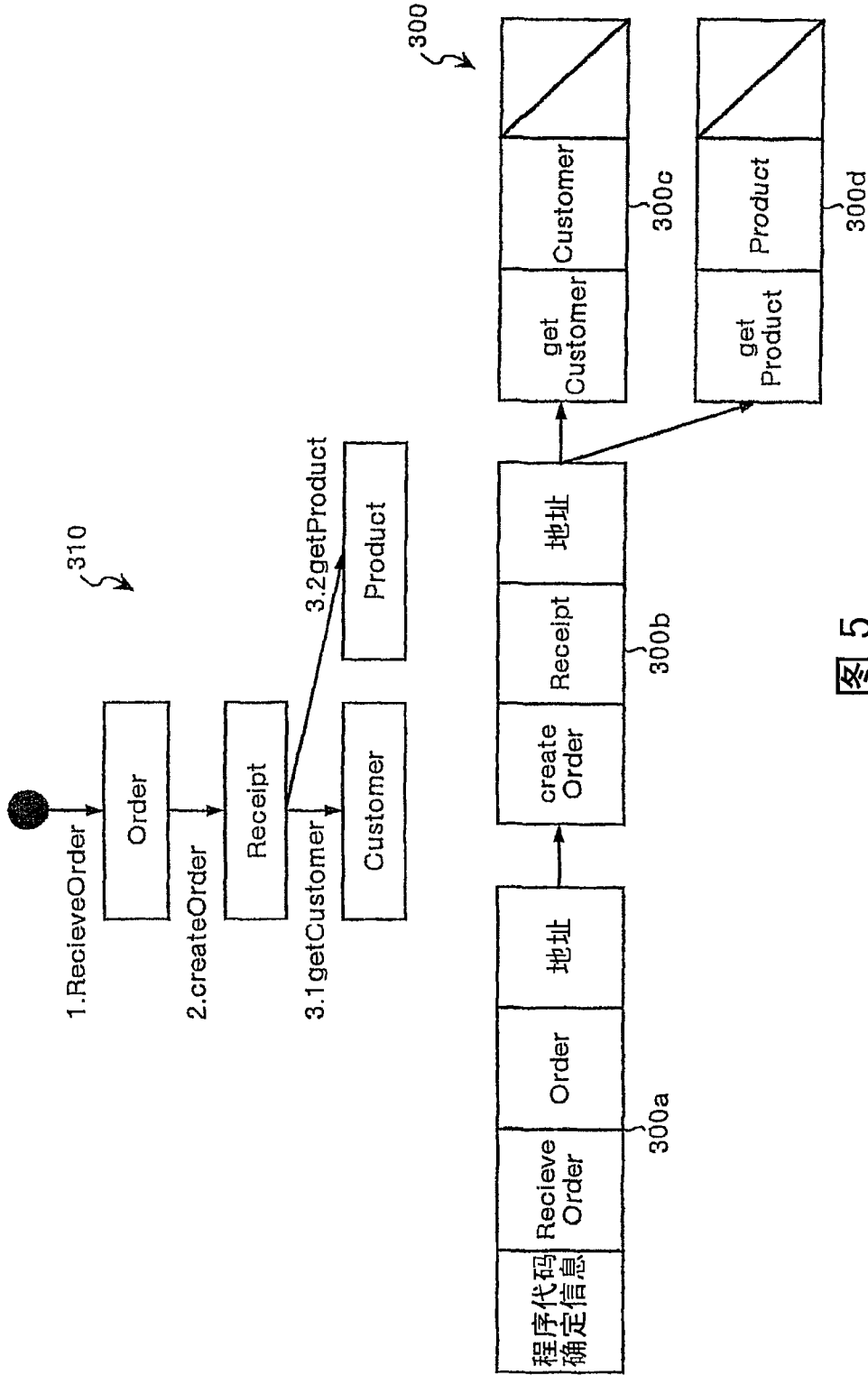


图 5

对应表

类名	实例名
Order	a
Receipt	b
Product	c
Customer	d

图 6

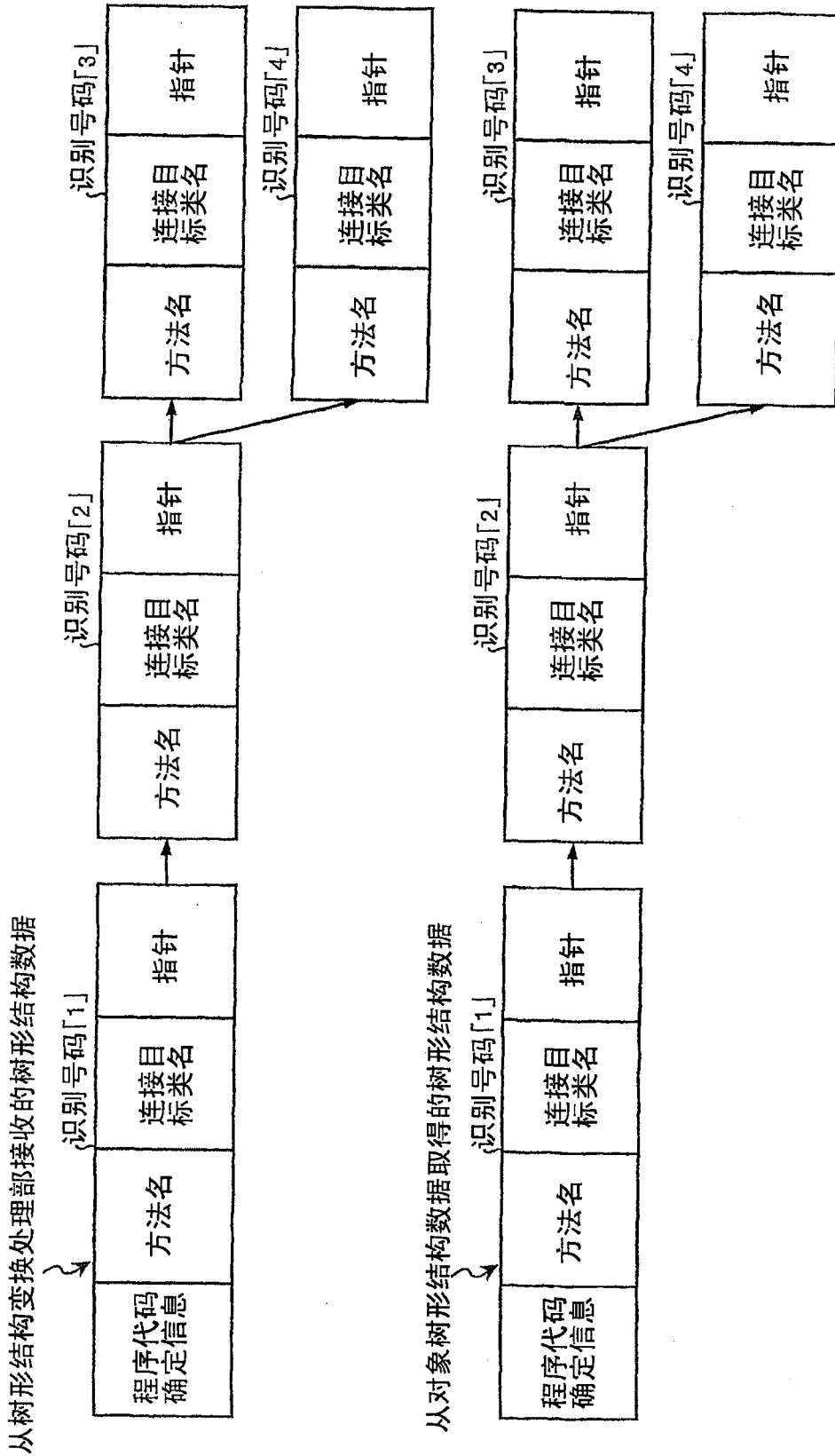


图 7

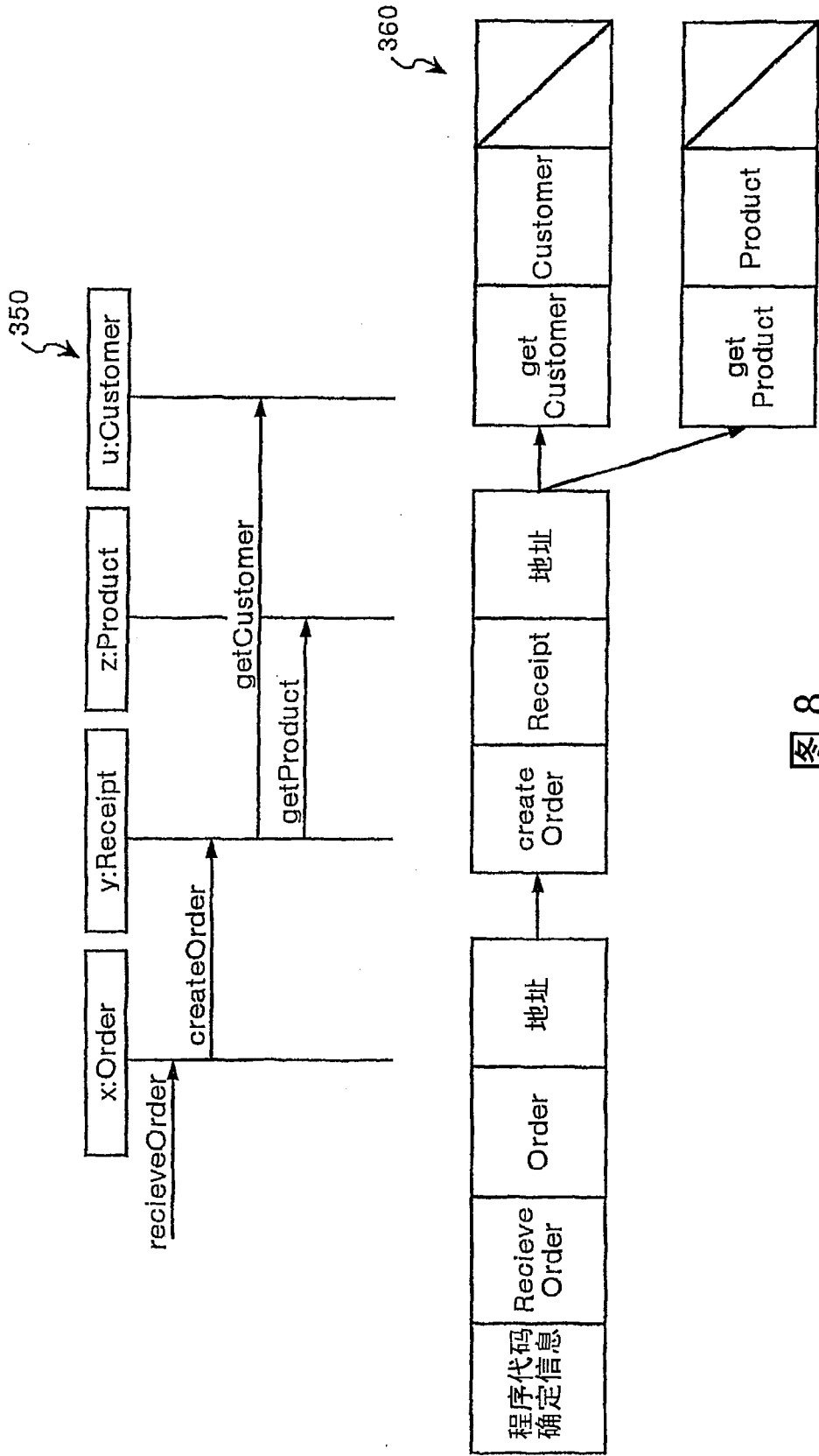


图 8



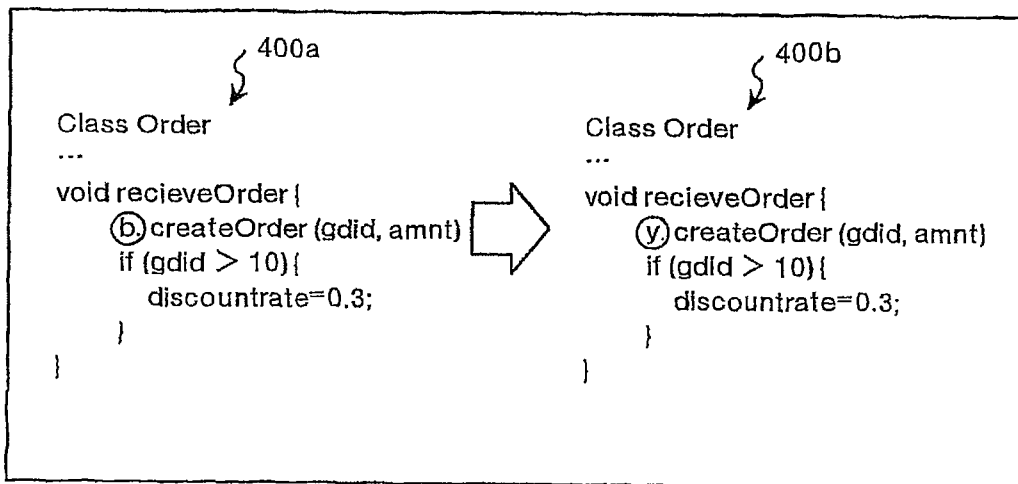


图 9

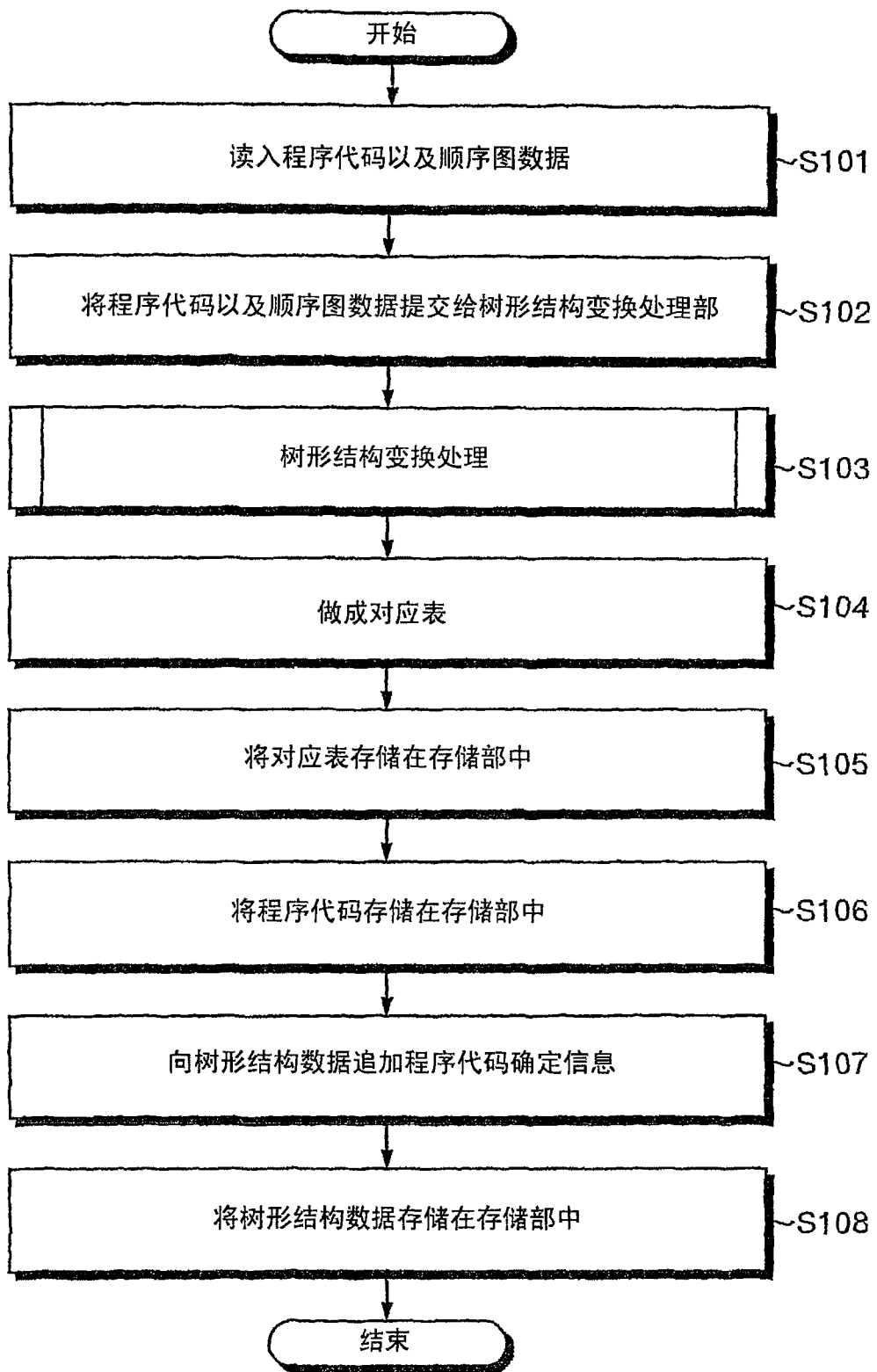


图 10

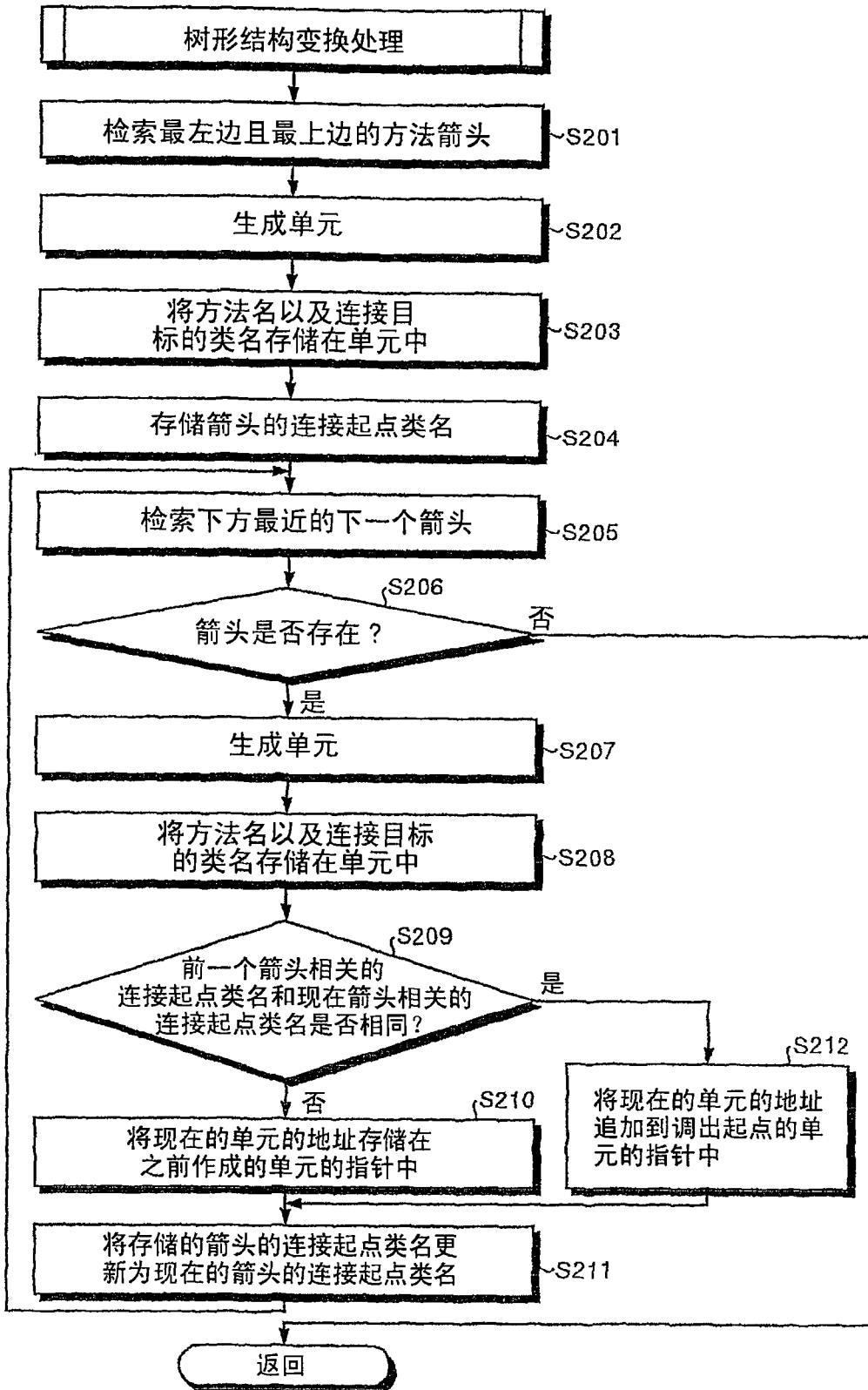


图 11

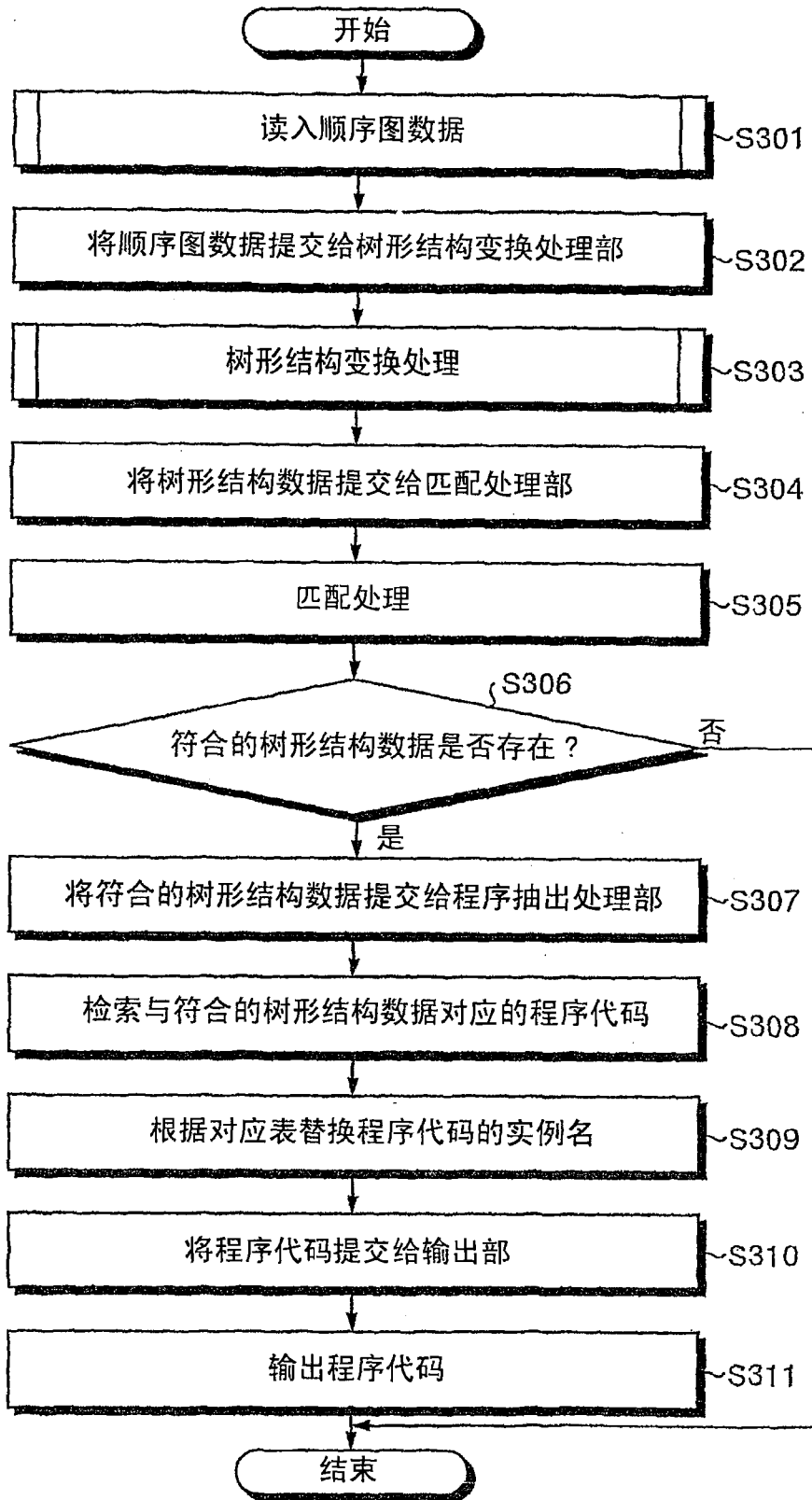


图 12

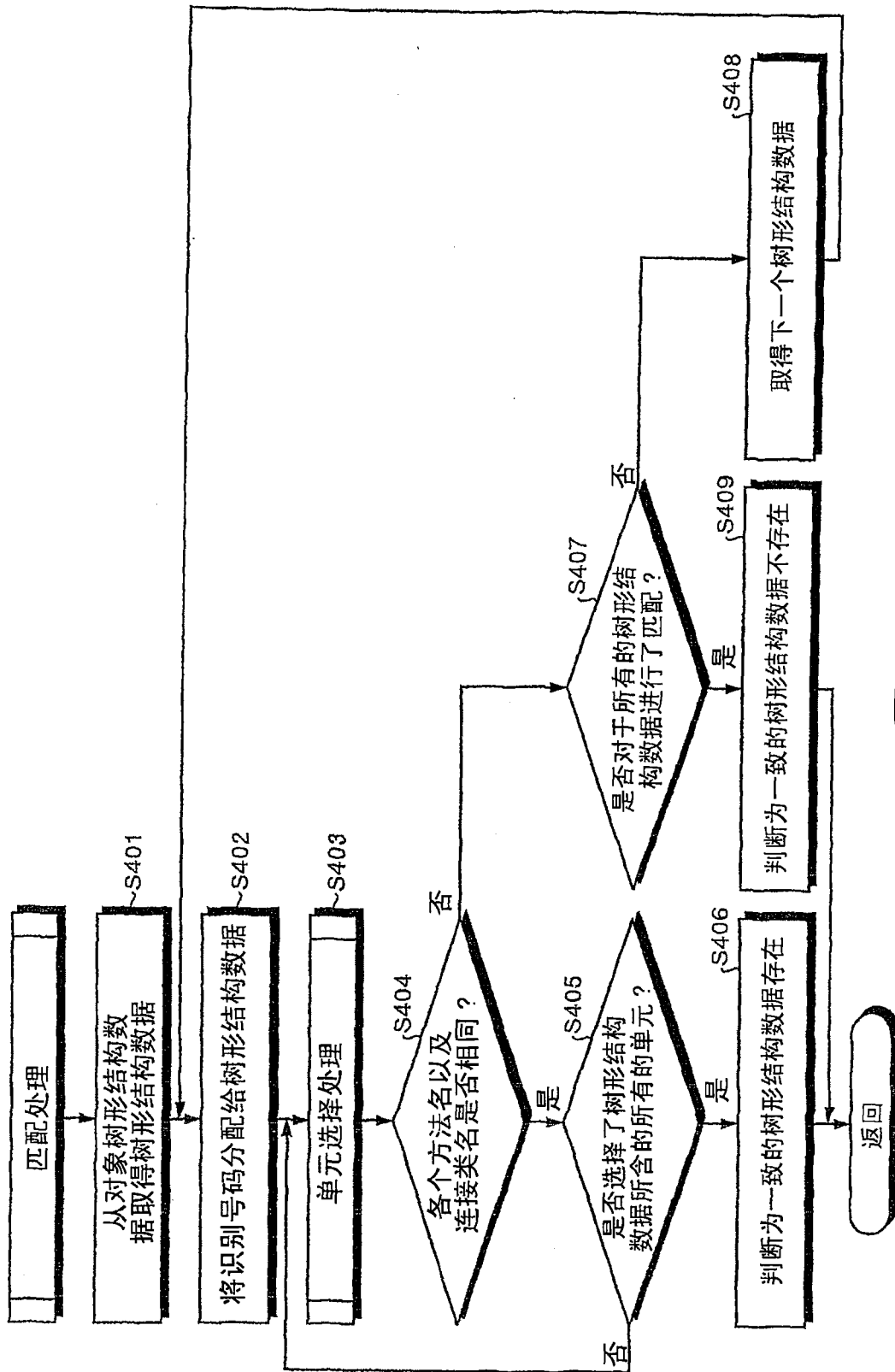


图 13

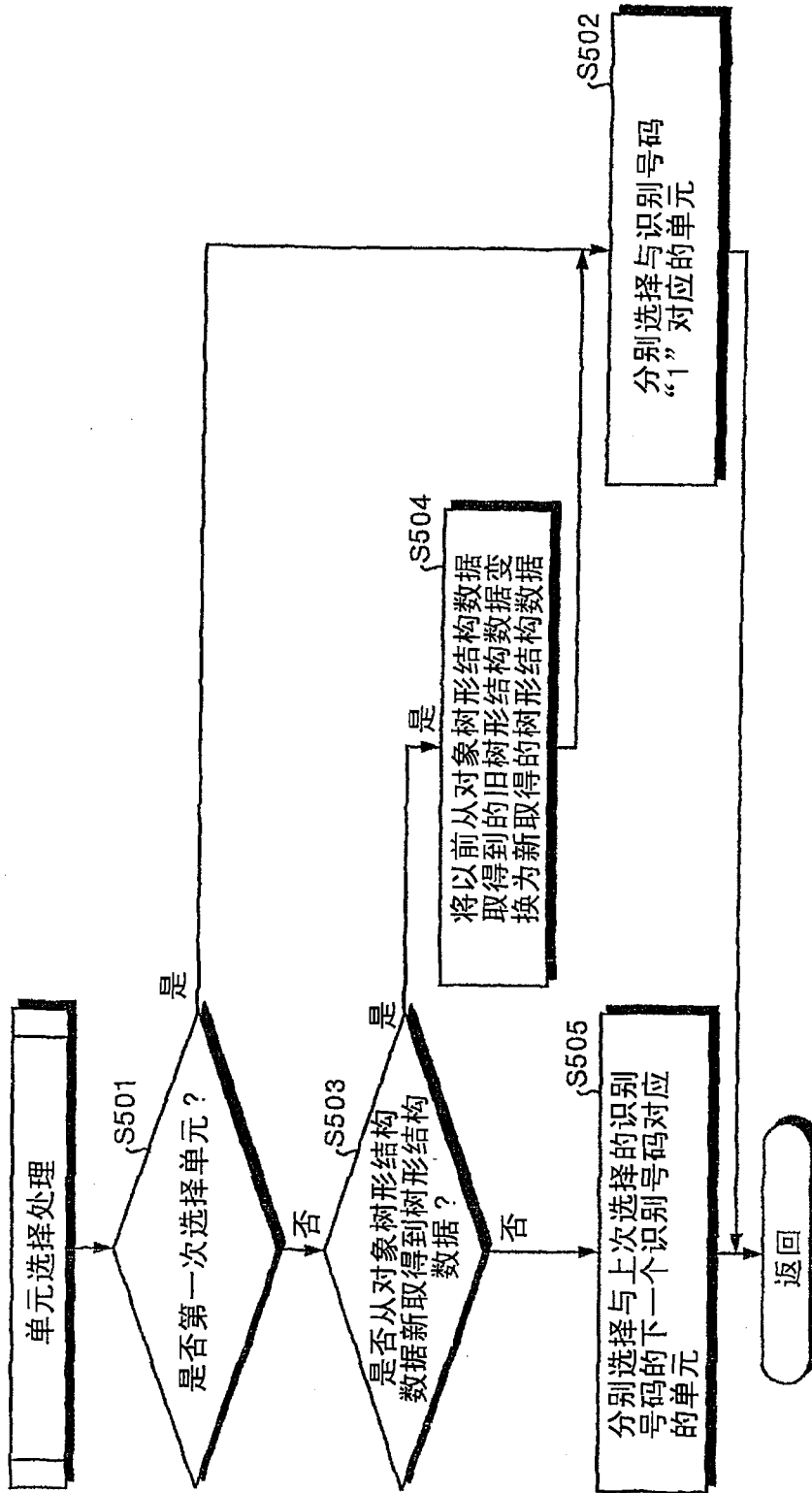


图 14

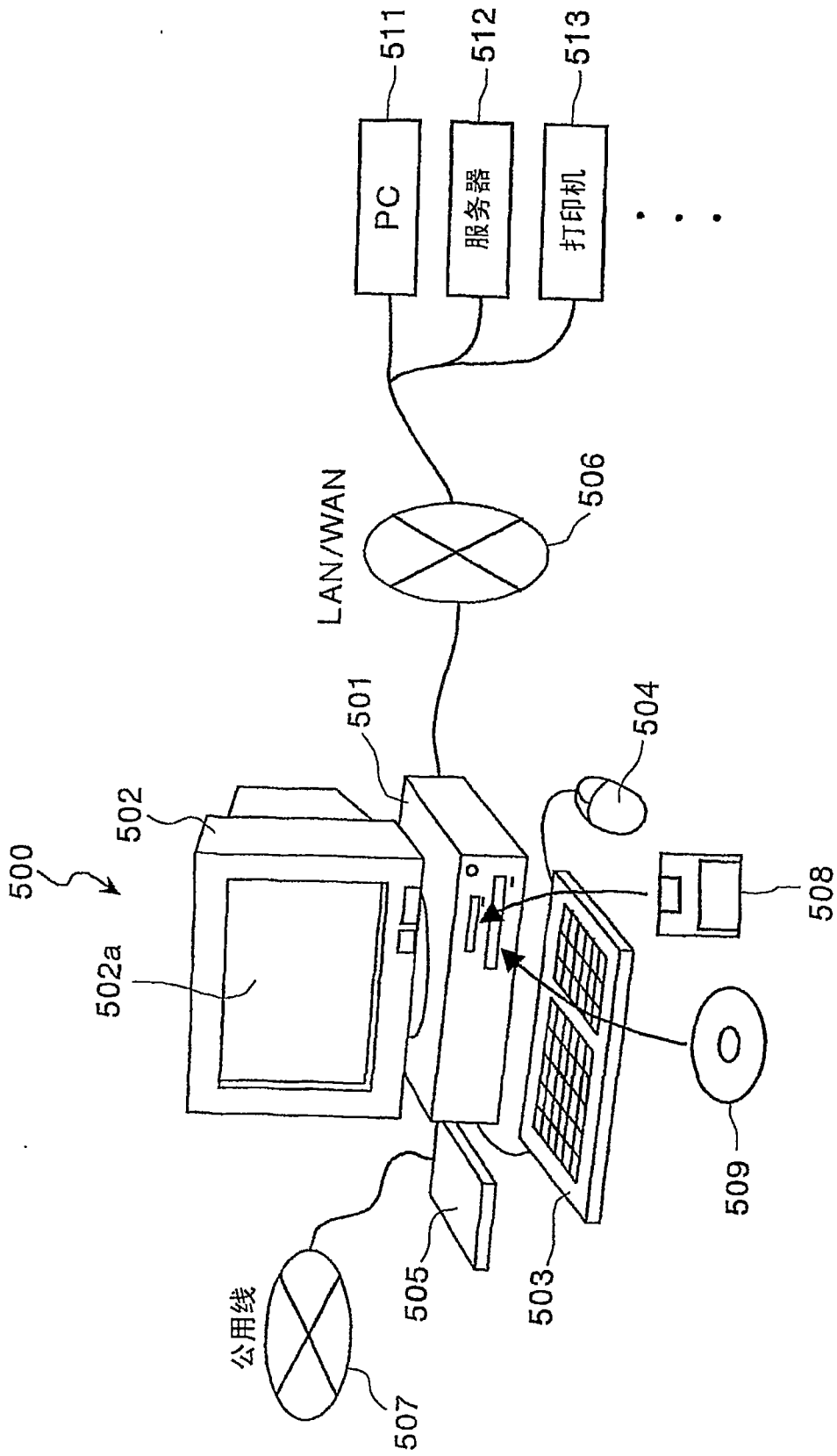


图 15

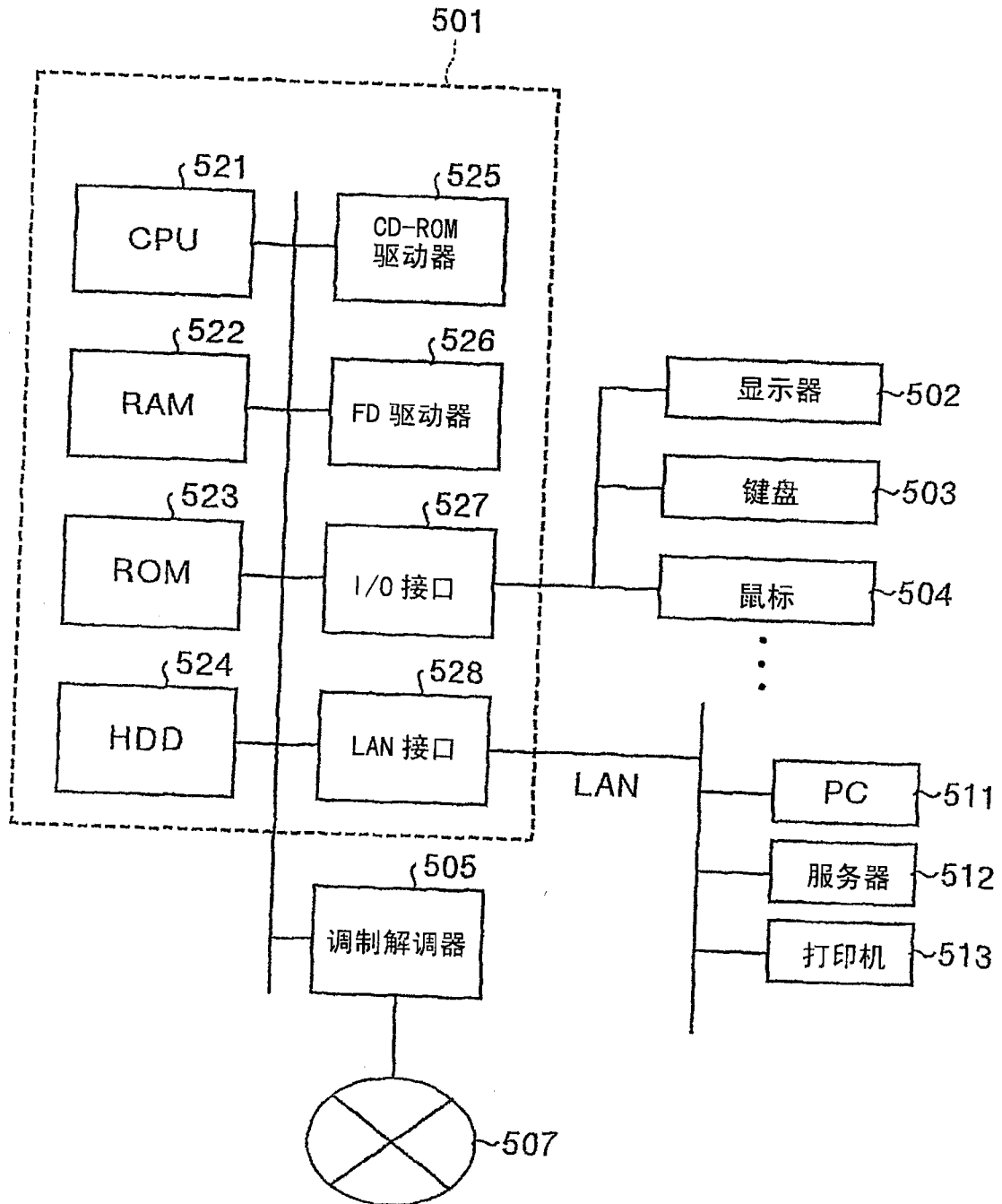


图 16