[19]中华人民共和国专利局

[51]Int.Cl[6]
G06F 9/40

[12] 发明专利申请公开说明书

[21] 申请号 94194917.6

[43]公开日 1997 年 2 月 12 日

[11] 公开号 CN 1142870A

[22]申请日 94.11.21

[30]优先权

[32]93.12.2 [33]US[31]08 / 161,229

[86]国际申请 PCT / US94 / 13567 94.11.21

[87]国际公布 WO95 / 16235 英 95.6.15

[85]进入国家阶段日期 96.7.29

[71]申请人 马克布伦德尔公司

地址 美国纽约州

[72]发明人 约翰·E·T·布拉安

[74]专利代理机构 永新专利商标代理有限公司

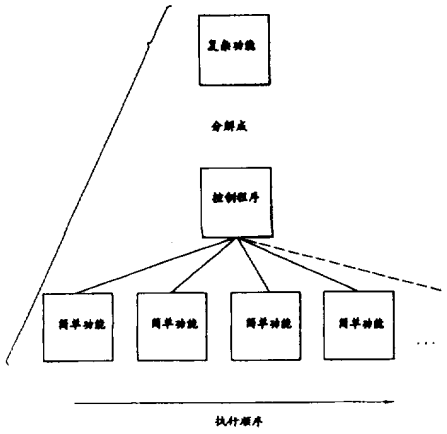
代理人 蹇 炜

权利要求书 8 页 说明书 14 页 附图页数 6 页

[54]发明名称 机器间通信的方法与过程及其程序制备的一般化方法

[57]摘要

一种计算机编程方法，用于将源消息翻译成目标消息。此翻译步骤是叙述于一数据库中，而数据库中储存了所有必需的指令与数据。控制程序具有响应源消息及访问数据库的功能。数据库指导控制程序以存取调用翻译方法程序的子例行程序；翻译方法程序存取源消息，将源消息翻译成目标消息；将消息送至控制程序，以告知翻译处理中的各步骤是否已成功地达成；然后响应控制程序，将完全翻译后的目的消息送至接收器。



权　利　要　求　书

1 、一种后配联计算机程序，　在对具有临时存储器的类型的计算机进行设定时作用于源消息，将一预定的源消息翻译成一预定的目标消息，此后配联程序包含:

(a) 提供一控制系统;
(b) 提供源消息至控制程序;
(c) 提供一数据库;
(d) 储存数据库中的数据与指令;
(e) 使控制程序访问数据库;
(f) 使控制程序响应储存于数据库中的数据与指令;
(g) 提供至少一翻译方法或子程序;
(h) 使控制程序响应储存于数据库中的指令，调用翻译方法程序;
(i) 使翻译方法程序响应数据库中的指令，翻译至少部分的接收消息; 及
(j) 响应储存于数据库中的指令与数据，限制控制程序响应源消息、访问数据库、及执行翻译方法程序。

2 、根据权利要求1 所述的后配联程序，其中该储存数据库中的数据与指令的步骤包括: 储存控制程序欲执行的翻译方法程序调用指令; 且提供翻译方法程序的步骤包括: 在至少一部分的源消息上提供一翻译方法程序用于操作。

3 、根据权利要求2 所述的后配联程序，其中提供指令与数据至数据库的步骤包括: 对于各应用方法程序提供一独特串的字符或函数名称: 提供一函数给控制程序以将函数名称转换成可执行的函数引用。

4 、根据权利要求3 所述的后配联程序，其中使控制程序响应数据库中这数据与指令的步骤包括: 在后配联程序开始时，使控制程序响应指令与数据。

5 、根据权利要求4 所述的后配联程序，其中在后配联程序开始时，令控制程序响应数据库中指令的步骤，使控制程序访问数据库以将函数名称从从数据库移除。

6 、根据权利要求5 所述的后配联程序，其中使控制程序响应数据库中的指令的步骤，在储存于数据库中的数据与指令的指导下，控制程序设计步骤的顺序。

7 、根据权利要求6 所述的后配联程序，其中储存数据库中的指令与数据的步骤包括：提供可能值范围的分类表及用于导找非演算翻译所使用的一般化机构表。

8 、根据权利要求7 所述的后配联程序，其中提供翻译方法程序的步骤包括提供许多翻译方法程序；执行翻译方法程序的步骤包括使控制程序存取数据库以得到一源数据的位置的引用、一目标数据的位置的引用（翻译后的源数据必须写至此目标数据），各源数据及目标数据的最大长度与用以查询数据库中指令所载任一表列的引用。

9 、根据权利要求8 所述的后配联程序，其中翻译的步骤包含：拷贝步骤（由源与目标数据决定辨识左或右）；将字符转换成整数数字格式（反的亦然）；将字符转换成浮点数字格式（反的亦然）；将字符转换成浮点数字格式（反的亦然）；将浮点转换成整数数字格式（反的亦然）；查输入数据档中表的值（输出结果作为字符、整数、或浮点）。

1 0 、根据权利要求8 所述的后配联程序，其中调用翻译方法程序的步骤包括：从数据库提供先前储存的指令与数据，包含提供一引用给翻译方法程序码；翻译方法程序所需要作为参数的数据及藉翻译方法程序从数据库回传至调用指令的数据。

1 1 、根据权利要求1 0 所述的后配联程序，其中提供一参考数据给翻译方法程序的步骤包含提供一储存于数据库中的函数名称。

1 2 、根据权利要求1 0 所述的后配联程序，其中使控制程序响应指令与数据的步骤，进一步包含控制程序从储存于数据库中的函数名称导出函数指针。

1 3 、根据权利要求1 2 所述的后配联程序，其中在使翻译方法程序翻译的步骤中，翻译方法程序访问数据库。

1 4 、根据权利要求1 0 所述的后配联程序，包括具有一翻译方法程序，将数据转回至控制程序，指出将源数据翻译成目标数据的成功或失败。

1 5 、根据权利要求1 3 所述的后配联程序，包括要求翻译方法的参数是固定的。

１６、根据权利要求1３所述的后配联程序，包括要求目标与源消息的结构是预先决定的。

１７、根据权利要求9所述的后配联程序，其中翻译步骤进一步包含：以右或左对齐来逐字节地拷贝；查分类表中的源值并设定目标成对应源值的值；查分类表中的源值并设定目标成对应源值的值，转换成短整数；查分类表中的源值并设定目标成对应源值的值，转换成整数；查分类表中的源值并设定目标成对应源值的值，转换成长整数；查分类表中的源值并设定目标成对应源值的值、转换成倍精度浮点数字；将字符（只包含数字字符、成千个可选择的定界标与空间）转换成对应的短整数值；将字符（只包含数字字符、成千个可选择的定界标与空间）转换成对应的整数值；将字符（只包含数字字符、成千个可选择的定界标与空间）转换成对应的长整数值；将字符（只包含数字字符、成千个可选择的定界标、可选择的十进位字符与空间）转换成对应的浮点值；将字符（只包含数字字符、成千个可选择的定界标、可选择的十进位字符与空间）转换成对应的倍准度浮点值；将短整数转换成字符字段；将整数转换成字符字段；将长整数转换成字符字段；将浮点数字转换成字符字段；将倍精度浮点数字转换成字符字段；将倍精度浮点数字转换成ＰＩＣＴＵＲＥ字段，如同界定于ＣＯＢＯＬ程序设计语言；将界定于ＣＯＢＯＬ程序设计语言中的ＰＩＣＴＵＲＥ字段转换成倍精度浮点数字；设定目标字段中的一任意值；将源值加至短整数目标领域；将源值加至长整数目标字段；将源值加至浮点目标字段；将源值加至倍精度浮点目标字段；从短整数目标字段减去源值；从整数目标字段减去源值；从长整数目标字段减去源值；从浮点目标字段减去源值；从倍精度浮点目标字段减去源值；将源值乘以短整数目标字段；将源值乘以整数目标字段；将源值乘以长整数目标字段；将源值乘以浮点目标字段；将源值乘以倍精度浮点目标字段；将源值除以短整数目标字段；将源值除以整数目标字段；将源值除以长整数目标字段；将源值除以浮点目标字段；将源值除以倍精度浮点目标领域；施加逻辑位元“与”操作于源字段与目标字段的间；施加逻辑位元“或”操作于源字段与目标字段的间；施加逻辑位元“或”操作于源字段与目标字段的间。

１８、一种后配联计算机程序方法用于制作程序，其中提供源刺激给后配联计算机程序，其中各刺激的型式包含一组预定数目的事件的型式及一组预定的数据结构的型式，且其中后配联计算机程序的型式可以调用至少一子程序（亦即一[应用方法程序]），此后配联计算机程序方法包含：

(a) 提供一控制系统；

(b) 提供源刺激至控制程序;

(c) 提供一数据库;

(d) 储存欲作用于数据库中的任意源刺激的结构数据与指令;

(e) 使控制程序访问数据库;

(f) 使控制程序响应储存于数据库中的数据与指令;

(g) 使控制程序响应数据库中的数据、指令、刺激，能够调用至少一应用方法程序; 及

(h) 响应储存于数据库中的指令与数据，限制控制程序响应源刺激、取存数据库。

1 9 、根据权利要求1 8 所述的后配联程序，进一步包含提供一组预定的应用方法程序。

2 0 、根据权利要求1 9 所述的后配联程序，进一步包含将全部预定的应用方法程序的参数储存于数据库中; 其中使控制程序响应源刺激的步骤包含: 在控制程序从数据库所发送的数据与指令的指导的下，控制程序将参数的预定结构传送至预定应用方法程序。

2 1 、根据权利要求2 0 所述的后配联程序，其中储存步骤包括将参数结构储存于数据库中。

2 2 、根据权利要求2 0 所述的后配联程序，其中控制程序传送参数的步骤进一步包含: 由控制程序所发送的数据及指令，可决定将调用那一种预定应用方法程序。

2 3 、根据权利要求2 0 所述的后配联程序，其中调用步骤包括提供参数给应用方法程序。

2 4 、根据权利要求2 0 所述的后配联程序，进一步包含提供包裹; 提供控制程序一种能力可调用至少一包裹; 使用包裹以调用应用方法程序或其它的包裹。

2 5 、根据权利要求2 0 所述的后配联程序，其中储存指令与数据于数据库中的步骤包括: 储存应用方法程序调用; 且控制程序响应得自数据库的信息与刺激，能够调用至少一预定应用方法程序。

2 6 、根据权利要求2 0 所述的后配联程序，其中提供指令与数据给数据库

的步骤包括：提供一独特串的字符或函数名称给应用方法程序或包裹；提供一函数给控制程序，以将函数名称转换成函数指针。

27、根据权利要求26所述的后配联程序，进一步包含提供装置，用于将刺激的预定数据翻译成应用方法程序与包裹的预定参数结构。

28、根据权利要求25所述的后配联程序，进一步包含使用控制程序，在储存于数据库中的数据与指令的指导下，来控制调用的顺序。

29、根据权利要求27所述的后配联程序，其中储存数据与指令于数据库中的步骤包括：提供一范围的可能值的分类表及一机构用于寻找非演算翻译的表。

30、根据权利要求25所述的后配联程序，其中提供应用方法程序的步骤包括：提供许多应用方法程序；执行应用方法程序的步骤包括：使控制程序访问数据库以得到参数结构用于应用方法程序、刺激数据的位置的参考数据、必须被写的应用方法参数的位置的参考数据，刺激结构的各单元的最大长度，刺激数据与参数结构的最大长度、及提出于数据库的指令中的查找表的参考数据。

31、根据权利要求28所述的后配联程序，其中翻译步骤包含：拷贝步骤（由应用方法的参数结构与刺激数据决定对齐左或右）；将字符转换成整数数字格式（反的亦然）；将字符转换成浮点数字格式（反的亦然）；将浮点转换成整数数字格式（反的亦然）；查输入数据档中表的值（输出结果作为字符、整数、或浮点）。

32、根据权利要求26所述的后配联程序，其中调用应用方法程序或包裹的步骤包括：从数据库提供先前储存的指令与数据，包含提供应用方法程序码或包裹码的参考数据；应用方法程序或包裹所需要的数据作为参数；由应用方法程序或包裹转回至控制程序的数据。

33、根据权利要求26所述的后配联程序，其中提供参考数据给应用方法程序的步骤包含提供一功能名称储存于数据库中。

34、根据权利要求26所述的后配联程序，包括具有各应用方法程序，提供指针给控制程序，其指针在指示各该应用方法程序的操作是成功或失败。

35、根据权利要求34所述的后配联程序，提供指针的步骤包括当控制程

序调用应用方法程序时，将指针从应用方法程序直接通过至控制程序。

36、根据权利要求34所述的后配联程序，提供指针的步骤包括当至少一包裹调用应用方法程序时，从应用方法程序经由包裹将指针通过至控制程序。

37、根据权利要求31所述的后配联程序，其中翻译步骤进一步包含：以右或左辨识来拷贝位元对位元；查分类表中的源值并设定目标成对应源值的值；查分类表中的源值并设定目标成对应源值的值，转换成短的整数；查分类表中的源值并设定目标成对应源值的值，转换成整数；查分类表中的源值并设定目标成对应源值的值，转换成长的整数；查分类表中的源值并设定目标成对应源值的值，转换成倍精度浮点数字；将字符（只包含数字字符、成千个可选择的定界标与空间）转换成对应的短整数值；将字符（只包含数字字符、成千个可选择的定界标、与空间）转换成对应的整数值；将字符（只包含数字字符、成千个可选择的定界标与空间）转换成对应的长整数值；将字符（只包含数字字符、成千个可选择的定界标、可选择的十进位字符与空间）转换成对应的浮点值；将字符（只包含数字字符、成千个可选择的定界标、可选择的十进位字符与空间）转换成对应的倍精度浮点值；将短整数转换成字符字段；将整数转换成字符字段；将长整数转换成字符字段；将浮点数字转换成字符字段；将倍精度浮点数字转换成字符字段；将倍精度浮点数字转换成PICTURE字段，如同界定于COBOL程序设计语言；将界定于COBOL程序设计语言中的PICTURE字段转换成倍精度浮点数字；设定目标字段中的一任意值；将源值加至短整数目标字段；将源值加至长整数目标字段；将源值加至浮点目标字段；将源值加至倍精度浮点目标字段；从短整数目标字段减去源值；从整数目标字段减去源值；从长整数目标字段减去源值；从浮点目标字段减去源值；从倍精度浮点目标字段减去源值；将源值乘以短整数目标字段；将源值乘以整数目标字段；将源值乘以长整数目标字段；将源值乘以浮点目标字段；将源值乘以倍精度浮点目标字段；将源值除以短整数目标字段；将源值除以整数目标字段；将源值除以长整数目标字段；将源值除以浮点目标字段将源值除以倍精度浮点目标字段；施加逻辑位元“与”操作于源字段与目标字段之间；施加逻辑位元“或”操作于源字段与源字段与目标字段之间；施加逻辑位元“异或”操作于来源字段与目标字段之间。

38、一种写入翻译程序的方法，将源消息翻译成一目标消息，此方法包含：

(a) 辨识目标消息的参数与特征；
(b) 辨识源消息的特征；
(c) 导出需求以将源消息翻译成目标消息；

(d) 提供一数据库;

(e) 对应将源消息翻译成目标消息所需的步骤，储存数据与指令于数据库中;

(f) 写入一翻译方法程序，使源消息的各种型式的单元转换成目标消息的各种型式的单元;

(g) 将数据库中的参考数据储存至翻译方法程序;

(h) 储存辨识欲被调用的翻译方法程序的指令于数据库中;

(i) 限制控制程序使得在全部时间可由数据库中的数据与指令来控制; 及

(j) 在接收源消息时，使控制程序从数据库得到数据库中的指令以指导控制程序，并调用预定的翻译方法子程序，以将源消息翻译成目标消息。

3 9 、根据权利要求3 8 所述的方法，进一步包含使控制程序将翻译后的源消息传送至接收器。

4 0 、根据权利要求3 9 所述的方法，其中辨识目标与源消息的参数和特征的步骤包括: 辨识目标消息与源消息的最简单的数据单元。

4 1 、根据权利要求4 0 所述的方法，其中导出翻译的步骤包括决定将各源消息的各简单单元翻译成目标消息的结果。

4 2 、根据权利要求4 1 所述的方法，其中导出翻译的步骤进一步包含决定由翻译程序所采取的翻译步骤。

4 3 、根据权利要求4 2 所述的方法，其中导出翻译的步骤进一步包含决定翻译的步骤的顺序。

4 4 、根据权利要求4 3 所述的方法，其中储存步骤包含: 储存翻译的步骤; 将数据与指令输入至表中; 输入翻译方法程序的位置及调用翻译方法程序的顺序。

4 5 、根据权利要求4 4 所述的方法，进一步包含在开始时使控制程序产生一区域于计算机的临时存储器中; 储存一表含有用于不同型式的源消息的辨识的数据; 储存指针于临时存储器区域中，用于辨识源消息中的源字段的位置; 储存查表于临时存储器区域中用于非演算翻译; 储存指令于临时存储器区域中，可辨识将调用那一种翻译方法程序; 储存操作顺序于临时存储器区域中，以此操作顺序来指导控制程序。

46、一种写入应用程序的方法，包含:

(a) 列出应用程序欲接收的全部刺激消息;

(b) 提供一列表指出各应用翻译方法程序欲执行的各操作及操作的顺序;

(c) 提供一数据库;

(d) 储存刺激于数据库的表中;

(e) 将执行数据库中的应用方法程序的指令储存作为一消息，此消息包含应用方法程序的参考数据及应用方法程序将使用的参数的引用;

(f) 决定将由应用程序所接收的各刺激，并将刺激命名作为一源消息;

(g) 决定欲执行的各应用方法程序的参数结构，并将其各命名作为一目标消息;

(h) 辨识各目标消息的特征及参数;

(i) 辨识源消息的特征;

(j) 导出规定以将源消息翻译成目标消息;

(k) 储存对应将源消息翻译成目标消息所需的步骤的指令与数据于数据库中;

(l) 得到各应用步骤的应用方法程序;

(m) 储存数据库中的指令，此指令辨识将调用那一个应用方法程序;

(n) 藉数据库中的指令使控制程序调用应用方法程序，并从源刺激单元产生与各应用方法程序有关的参数;

(o) 限制控制程序使得在全部时间可由数据库中的数据与指令来控制；及

(p) 在接收源消息时，使控制程序调用数据库；数据库中的指令导引控制程序，并调用预定的应用方法程序以将源消息翻译成目标消息，且提供目标消息作为至少完整程序的一部分。

47、根据权利要求46所述的方法，其中得到步骤包含写入一应用方法程序。

48、根据权利要求46所述的方法，其中得到步骤包含拷贝一应用方法程序。

说　　明　　书

机器间通信的方法与过程及其程序制备的一般化方法

本发明领域

本发明属于计算机程序建构，尤指数据翻译程序建构的领域。特别是关于一种后配联处理（l a t e　b i d i n g　p r o c e s s），用于提供一种有效的系统，将信息由一计算机组合并翻译至另一计算机。

现有技术的叙述

计算机系统及其所使用的程序在过去三十五年来已快速地发展。最早期的自动方法是大规模的分批数据处理工作，例如薪水册的制作。在找出各项问题并加以解决的际，产生了计算机程序的本体。然而，工业的本质为：新的程序最好是现存解决方式的改良。如此导致建造出许多与现存系统无关的新系统。目前独立系统的数量已大幅度地成长，很少注意到数据在不同竞争程序间的协调。

大量的竞争程序已导致软件建构技术的发展，而允许系统的重建更有效率。此种技术的发展产生了1 9 8 0 年代早期的第四代语言（4 G L）。虽然4 G L受到普遍预期将使置换此种系统方便又快速，可以每四或五年经常地更换主系统，但是这并没有发生。

当程序改良时，必须更换整个可执行的程序。导致此结果的原因是，程序产生的过程，采用许多手写源码的片断而编译成各个目的码，然后再由这些物件建立起单一的可执行程序。结果，任何片段源码中的任何改变都会在可执行的程序中产生重大的变动而可能引起副作用。在重写的后，程序必须接受回归测试，在测试中整个程序被重新测试以侦测出任何的副作用。所遭遇到的另一困难是将一版本的程序替换成另一程序。如此引起明显的时间降低与人员的重新训练。必须加上测试所需的大量的延迟。欲改善此问题的一个尝试，使用电脑辅助软件工程工具作为码产生器。这些技术的目的，对于不同的成功度，在加速程序的制作。然而，它们并没有解决改良一程序或产生一新程序所必须采取的多项步骤的主要困难。现有技术中制备一程序或修正此程序的步骤请参见图1，其中先“改变需求”，接着“设计软件改变”，接着“设计”。然后“原始程序”、与“设计”、

接着“改变程序”、“改变后的程序”、“系统测试”、“接受新程序”、“更换原始程序”、“新程序”。

程序制作的另一种方法通常称为面向对象（“００”）。支持此种程序制作方法的一般理论为：对任何型式计算机系统，有一组非琐碎的操作及使用于许多情形的相关信息的辨识。例如，以使某些银行操作自动化的计算机系统中，会有处理金钱数量的经常性需求。这些处理并非计算机的基本操作的一部份，它们是由许多计算操作所组成，这些计算操作形成许多较大的演算法的共同部份。在００环境中，这些部分演算法及其相关的数据结构，被视为“对象”。此种技术是已知制作用于一般重覆性工作次程序的实际应用的延伸。００程序语言包括可使此分割程序的方法能比先前存在的语言更简单且更有效的结构。００技术的整个效果是与码产生器类似：在改良“原始程序”制作阶段（图1）。

系统测试与程序更换所引起的困难是由于计算机源码与可执行映像的间的差异。源码是由程序员或码产生器所产生的。通常是写成人类可读取的语言（即例如ＣＯＢＯＬ，ＦＯＲＴＲＡＮ７７，ＰＲＯＬＯＧ等）。这些语言各具非常正式的文法、语法的规则。使用源码将演算法从一1 计改写成编译程序能解译的型式。编译程序为将源码转换成中间型式的目的码的程序。目的码是界于人类可读取的源码与可执行映像的间的中间型式，此可执行映像可实际地在计算机中执行。最后的阶段是称为连接程序的程序的执行，此连接程序将一或多个片段的目的码转换成可由计算机执行的型式。此种程序制备的已知方法的流程表示于图2。可执行映像是在计算机上执行程序的唯一型式，单独决定了程序的行为。源码是可由程序员或码产生器处理的唯一型式。因为程序的可执行映像是非常复杂的，会导致前面所提及副作用。在可执行映像中常可能发生不可预见的副作用，这些副作用是源码的程序员或码产生器所不要的。

更换系统时由包含于系统中的数据转换的逻辑、植入新的软件、使用者的再教育所引起的整体问题，意谓着这些系统将继续存在一段相当长的时间。此意谓着这些系统必须使用于多用户连网系统，使得这些系统所包含的数据可以为后来发展的系统所使用。

另一个困难在于不同系统的间的相互作用。多系统的间的连接通常是手动完成的。例如，来自一计算机系统的印刷记录被人工输入至第二个系统。这是慢而冗长的，因此是无效率而昂贵的。另一种已知的方法是放置一新的程序于现存的系统的间，作用为中间程序。在上述两种情形中，都需要程序设计。

发明概述

本发明的目的是提供一种计算机化方法，可有效地将源消息翻译成目标消息。

本发明的更另一个目的是提供一种计算机程序，其中可以改变程序而不需要重写程序的整个源码。

本发明的进一步目的是提供一种方法，通过储存及调用基本编程单元并在程序的执行时间组合所使用的这些单元来制作程序。

依据本发明的目的，提供一种后配联计算机程序方法，在对具有临时存储器的类型的计算机进行设定时作用于源消息，其中程序翻译一预定源的消息成一预定的目标消息，此后配联程序包含：

(a) 提供一控制程序;
(b) 提供源消息至控制程序;
(c) 提供一数据库;
(d) 储存数据库中的数据与指令;
(e) 使控制程序访问数据库;
(f) 使控制程序响应储存于数据库中的数据与指令;
(g) 提供至少一翻译方法或子程序;
(h) 使控制程序响应储存于数据库中的指令，调用翻译方法程序;
(i) 使翻译方法程序响应数据库中的指令，翻译至少部份的接收消息；及
(j) 响应储存于数据库中的指令与数据，限制控制程序响应源消息、取存数据库、及执行翻译方法程序。

在本发明的另一个实施例中，提供一种后配联计算机程序方法用于制作程序，其中提供源刺激给后配联计算机程序，其中各刺激的型式包含一组预定数目的事件的型式及一组预定的数据结构的型式，后配联计算机程序的型式可以调用至少一子程序（亦即－“应用方法程序”），此后配联计算机程序方法包含：

(a) 提供一控制系统;
(b) 提供源刺激至控制程序;
(c) 提供一数据库;
(d) 储存欲作用于数据库中的任意源刺激的结构、数据与指令;
(e) 使控制程序访问数据库;

(f) 使控制程序响应储存于数据库中的数据与指令;

(g) 使控制程序响应数据库中的数据、指令、刺激，能够调用至少一应用方法程序；及

(h) 响应储存于数据库中的指令与数据，限制控制程序响应源刺激、取存数据库。

在本发明的更另一个实施例中，提供一种写入翻译程序的方法，将源消息翻译成一目标消息，此方法包含:

(a) 辨识目标消息的参数与特征;

(b) 辨识源消息的特征;

(c) 导出需求以将源消息翻译成目标消息;

(d) 提供一数据库;

(e) 对应将源消息翻译成目标消息所需的步骤，储存数据库数据与指令;

(f) 写入一翻译方法程序，使源消息的各种型式的单元转换成目标消息的各种型式的单元;

(g) 将数据库中的参考数据储存至翻译方法程序;

(h) 储存数据库中辨识欲被调用的翻译方法程序的指令;

(i) 限制控制程序使得在全部时间可由数据库中的数据与指令来控制；及

(j) 在接收源消息时，使控制程序从数据库得到数据库中的指令以指导控制程序，并调用预定的翻译方法以将源消息翻译成目标消息。

在本发明的更另外一个实施例中，提供一种写入应用程序的方法，包含:

(a) 列出应用程序欲接收的全部刺激消息;

(b) 提供一列表指出各应用翻译方法程序欲执行的各操作及操作的顺序;

(c) 提供一数据库;

(d) 储存刺激于数据库的表中;

(e) 将执行数据库中的应用方法程序的指令储存作为一消息，此消息包含应用方法程序的参考数据及应用方法程序将使用的参数的参考数据;

(f) 决定将由应用程序所接收的各刺激，并将刺激命名作为一源消息;

(g) 决定欲执行的各应用方法程序的参数结构，并将其各命名作为一目标消息;

(h) 辨识各目标消息的特征及参数;

(i) 辨识源消息的特征;

(j) 导出需求以将源消息翻译成目标消息;

(k) 储存对应将源消息翻译成目标消息所需的步骤的指令与数据于数据库中;

(l) 得到各应用步骤的应用方法程序;

(m) 储存数据库中的指令，此指令辨识将调用那一个应用方法程序;

(n) 数据库中的指令使控制程序调用应用方法程序，并从源刺激的单元产生与各应用方法程序有关的参数;

(o) 限制控制程序使得在全部时间可由数据库中的数据与指令来控制；及

(p) 在接收源消息时，使控制程序调用数据库；数据库中的指令导引控制程序，并调用预定的应用方法程序以将源消息翻译成目标消息，且提供目标消息作为至少完整程序的一部份。

附图的简要叙述

图1 是已知技术中制备一程序所需的步骤的流程图;

图2 是程序制备的步骤的流程图;

图3 是一图形指出一典型的程序的分解;

图4 是一图形指出一翻译程序;

图5 是一方法程序的图形;

图6 是一方法程序的数据结构的图形;

图7 是一图形，指出从一控制程序至一方法的请求及对请求的响应。

较佳实施例的叙述

对于机构有一需求，希望能帮助计算机系统的快速积分，第二个需求是更快速地发展及改变系统。单一的解决方式能够提供这两个问题的解答。

可以通过“结合时间”的已知的观念，精确地预测制备一新的程序或重写一程序所需的时间。欲分析计算机程序操作，需使用结合时间的观念。结合时间叙述产生程序的点，在此点其行为是固定的。例如，若一程序包含一行:

X： =3 +4

(亦即将变数X 设为3 与4 的和)

那么执行此语句所产生的X 的值将永远是7 。所以此语句的“结合时间”的行为是编译程序的时间。此将是最早的可能的结合时间。

现在考虑另一个语句:

X: =a +b

(亦即将变数X 给定为变数a 与b 的和的值) 。

此外, 将此语句放入一子程序中:

```
subroutine (a、b)
X: =a y +b;
return x;
```

此语句的行为是现在无法从此演算式来决定X 的值, 因为其是视变数a 与b 的值而定。变数a 与b 是由“调用”子程序的程序所拥有。此调用/ 响应关系在编译时没有决定。当程序被链结进入可执行的形式时, 才将其决定。结果, “结合时间”被移回至“链结编辑”时间。

作为最后一个例子, 想像一程序包含语句:

X: =a +b

此时程序是以一种方式构成, 使得a 与b 的值是由使用者在一终端机设定而提供的。在此情形中, 没有决定程式的行为直到程序运转时。“结合时间”现在为“执行时间”。由于程序可以正确地执行动作, 这些动作并非由设计者所明确地想出的, 所以此结果是有用的。任意一设计例如一相加程序, 在可相加在一起的两个数目 (a 、b) 的每一个可能组合, 不需经过明确的设计的整个处理, 但程序仍能将任意两个数目相加。

有后迟“结合时间”的程序的优点为: 它们不需要重新编译或重新链结以提供新功能。程序改变的实现与使用此程序方法的新程序, 将系统的测试时间减少至新功能所请求的测试。旧功能并不受影响, 因为可执行映像没有改变。由于可执行映像没有改变, 在运转系统中更换时并不会造成不便。

全部的计算机程序表示演算法。这些演算法包含依顺序的指令或简单的功能。结果, 即使是复杂的计算机程序亦可藉由将其分解成相关的单元 (图3) 而加以叙述。并没有界定指令的种类, 但是这些指令通常是适于手动工作的小单元处理。实际上, 在做成计算机系统时, “指令”的形式通常是由手动工作来界定, 然后使用一适当的程序语言来表示。已知这些功能最好被植入作为子程序, 这些子程序可依类似的指令需求被重新使用。

后配联程序的特征为当它们执行时它们需要信息，此信息告诉它们对于一特殊刺激如何反应。在执行时间结合的叙述中，藉由使用者在终端机的设定来提供信息。很清楚在大部分的环境中，此一情形将不被接受。必须使用另一个机构以提供所需的信息。其中一个此种机构是欲以数据库的形式来保持设定数据。此数据库包含对于一组环境设定一后配联程序所使用的信息。设定资料库的改变会改变程序的行为。

数据库包含对于一特殊的刺激的欲执行的处理步骤的引用。由于演算法的定义是写成处理步骤，可执行演算的操作能叙述于后配联程序中，此程序是由数据库写成，其包含适当的预先存在的程序码，且可以执行计算机所能执行的任何处理。所以，后配联程序可以执行用途设计程序所能完成的工作。

将许多计算机系统连接在一起的工作，可减少成连接成对的系统的个别工作。连接器必须完成三项个别的工作。

这些工作为:

1 、使来自送出系统的数据对接收者是有意义的;

2 、建立一设备将计算机可读取的电子信息从第一系统通过至第二系统；及

3 、重建信息成接收者能接受的形式。

第一项工作是由有关的系统的检查而完成。第二项工作是由适当的电子传输协定所完成，此传输协定是由所连接的计算装置的特性来决定。以计算机系统来解决这些工作是不适当的。此两种工作需要视情形而定的良好的建立技术。第三种工作认定了不同计算机系统以不同形式表达信息。已知即使是在由相同的制造商所提供的系统中，真实世界中的叙述物体的数据结构在不同的系统中是不同的，即使所叙述的真实世界物体是相同的。植入系统中的延迟与复杂性的主要来源，是与执行需要的数据结构转换所需要的程序的制作有关。

有了此一般的背景，现在我们考虑后配联程序设计方法的实施。

所叙述的后配联程序是由两个部分构成：控制程序与子程序，它们执行需要的程序步骤。控制程序的目的是要以相关的数据来接收输入刺激，并导出欲采取的适当动作。此动作的形式为一列的欲调用的子程序，其是依被调用的顺序。然后控制程序以所要的顺序来调用需要的子程序，检查各调用是否成功。

在本发明的较佳实施例中，程序可由任何语言写成；最好是写成C 程序设计

语言。控制程序从数据库取出其指令，并将它们保留于计算机的存储器中，形式为阵列的数据结构（数据结构的取出进入存储器结构并不是重要的，但是能使程序非常快速地执行程序）。其中一个资料阵列是由全部的调用子程序所组成。这些数据结构包含一可执行的函数引用（在C 语言中它是一“函数指针”），对于子程序而言它是程序执行子程序所需要的引用。此函数指针无法保持于数据库中。对于全部所需要的子程序而言欲决定函数指针，数据库包含一独特串的字符用于各可用的子程序（其函数名称并实际地被编码进入各子程序）。控制程序包含可将这些函数名称转换成函数指针的功能。当计算机首先开始执行时，此操作是由控制系统所执行，对于程序的各刺激不需要执行此操作。

在控制程序中没有发生应用处理。其反应能力是完全与方法的选定、顺序与执行有关。控制程序从数据库取出其需要的信息，此数据库为了叙述可称为“中间数据库”。中间数据库中的改变将反映于程序的行为的改变。此处理是表示于图7 中。顺序是在中间数据库的方向下由控制程序所控制。

根据这些原理，可以不改变控制程序下来修改及制作新的程序。此观念可以使用于制作程序的方法及将信息源翻译成另一信息源的翻译程序。其理由为一旦控制程序建立基本步骤，一格式的数据翻译成另一格式的数据的操作是包含于方法及与方法共同操作的数据库结构中。

翻译程序（图4 ）是由以下方式操作：一旦得到欲翻译的数据（亦即源数据），其必须被辨识。此表示翻译程序必须辨识已接收到那一组的可能输入消息。此辨识决定一组输出结构（亦即目标数据），其必须被运送至接收器。此组可以包含零、1 、或许多输出结构。此翻译可以被分解成许多较小步骤的翻译，各应用于源数据中的简单值。

选择分解使得原始结构的各子部份的数据型式是由单一型式的计算机可读取数据（亦即，整数、浮点数目、位元图案或字符串）所做成。结果，数据型式的间的自由转换能力可以解决所需要的许多低阶翻译。检查不是纯构造的转换，可以看出主要部分并非能由演算法解决。这些通常是任意本文（a r b i t r a r y t e x t ）从一值转换成另一值。其中一个例子是货币码从一任意形式（例如R e u t e r s 所界定）转换成另一形式（例如S . W . I . F . T . 所界定）。数据的语意值没有改变，源数据与目标数据代表货币，但是值的间没有演算关系。此种型式的转换的解决方式是提供可能的源值与对应的目标值的范围的分类表，及一般化机构用于搜寻此表并将目标值取代成目标结构。

所以植入翻译程序所需的一组子程序含有一组结构转换副程序。执行功能如:

拷贝（将左或右作适当的对齐）；
将字符转换成整数数字格式（反的亦然）；
将字符转换成浮点数字格式（反的亦然）；
将浮点转换成整数数字格式（反的亦然）；
查输入数据档中表的值（输出结果作为字符、整数、或浮点）。

另一列的功能是附于附录中。当被调用时，各副程式被通过有：必须作为输入的数据的参考数据、其输出必须被写入的位置的参考数据、输入与输出的最大可允许长度、所使用的查表（如果有的话）的参考数据。　　所以，翻译程序能够执行任何可想到的数据翻译，在数据结构的参数的中，此参数可由C 程序设计语言叙述，此数据翻译能够叙述表示于现代数位计算机上的任何数据结构。

调用子程序的动作需要调用程序具有关于子程序的三段信息:

子程序码的参考数据（通常是函数名称或函数指针）；
子程序的参数所需要的数据;
由子程序送回至调用者的数据。

欲使控制程序能表示任何演算法，全部这三个特征必须是常数或是与控制程序的调用无关。副程序参考资料在整个前述的函数指针机构中是独立的。在翻译程序中，如前所述，对于全部可想的翻译子程序，参数数据为常数。由子程序所传回的数据被限制成一数值，指出成功或失败。

这些限制，此组的固定参数及固定的传回值，叙述特殊次级的子程序的规定，其作为翻译方法是已知的。这些子程序能够被组合成任何可想到的翻译程序。

翻译程序使用不同型式的方法、辨识方法，以执行输入消息的辨识。这些辨识方法接收所接收到的消息的位置、欲检查的输入消息的区域的偏移及长度（字节）、一列源消息的位置，作为它们的参数，其包括用于辨识的任意数据。辨识方法将一值传回至控制程序，表示已接收到输入消息型式。

在辨识方法中可执行的演算法的型式包括：对完全已知的数据值的比较、对部份已知的数据值的比较（使用“一般性的表示式”）、或对数据型式的比较。

对于全部的翻译活动，翻译程序采用翻译方法。欲延伸控制程序与副程序的处理模式，以包含一般用途的程序，需要延伸可以执行的子程序的范围。能够使用并非特地为此环境而建造的子程序是特别有用的。重新使用在其它程序中所使用的良好建立的码的能力，能使利用控制程序与子程序架构的程序可以做成非常地快速用于新的应用。

一般用途的程序设计的延伸是藉由移除子程序所使用的参数数据的限制而达成。已知使用于此环境中的子程序可作为一应用方法。应用方法是固定片段的程序码，可由以下三种方式的其中的一来得到:

1 、它们可以是手造的。

2 、它们可以是从现存的系统取出预先存在的老式功能而建成。

3 、它们可以是整个预先存在的系统。

藉由以相同的方式来处理全部的方法，控制程序可以获得其普遍性。存在于一环境中的方法是表示于图5 。一方法执行一演算法或部份的演算法。执行数据的演算法。此方法从以下得到数据:

由方法控制其存取的数据库;

与其它应用方法共用的数据（全局存）;

参数。

全部此三种型式的数据存取是经由数据结构来执行，此数据结构完全被控制于应用方法中。此允许方法可以彼此完全独立，除了有特别允许，藉着包括共用的资料结构。方法永远是与控制程序无关的。从数据库或经由共用的数据结构得到数据的应用方法所使用的机构，是固定为其内部构造的一部分。得到数据的这些机构是完全与控制程序无关的。然而，控制程序必须提供参数数据给应用方法，使得可以自由地重新地使用此参数资料。

欲使控制程序可以如此做，使用一翻译程序。当一刺激到达控制程序时，控制程序调用其内部翻译程序。翻译程序对于欲被执行的各应用方法制作一数据结构。数据结构取图6 所示的形式。藉由控制程序的延伸而使用此结构，以调用参考方法中的方法，通过参数于剩余的结构中。

此技术允许控制程序可执行以一般计算机语言写成的任何子程序。由于控制程序是以C 写成，此意谓着可使用能由C 调用的任何语言。

仍有某些情形，调用子程序的处理是不当的，或者其它的语言或实现限制会

避免控制程序直接存取子程序。在此情形中，使用一“包裹（wrapper）”程序（或一系列的包裹）。“包裹”是一种子程序，其目的是能被控制程序所调用，且能够调用所需要的应用方法程序。对于应用方法的构造的此小延伸能使软件开发者可利用任何可想到的片段软件作为应用方法。

制作程序的步骤包括一个别的步骤，以决定翻译程序的需要并了解将执行的工作的范围。此需要决定计算机将接收的一列输入消息。一旦组合了一列的可能消息的后，必须决定只根据结构与内容而作区别的机构。下一个步骤是对于各输入消息决定翻译程序的反应。换句话说，响应输入消息或刺激，翻译程序无法制作输出消息（亦即忽略输入读息）或制作一消息或制作许多消息。数目将是已知的。

下一个步骤是藉由辨识构成各目标消息的结构的简单数据单元来分解程序步骤。如同先前的步骤，其被手动地执行。然后接着是构成原始结构的最简单的单元的辨识，此单元对应源消息中的结果单元。对于源所需要的各原始结构简单单元，接着决定需要何种转换以产生适当的结果单元。如果顺序有任何需求，其必须列入考虑。例如，如果取平均值，在除的前先完成相加是重要的。在其它全部情形中，任意的顺序是可接受的。此分析的结果是一组转换指令，包含原始数据单元（亦即第一结构中的字段）、结果数据单元（亦即第二结构中的字段）及其间所需要的翻译方法。各构成中间数据库的XLAT细部表中的一列，其可以设定在翻译细部表的旁边作为XLAT-DETAILS。

然后数据被输入中间数据库中的表中。翻译程序的存在被记录在位置表中，在这里亦记录欲被调用的辨识方法，以区别输入消息与欲由辨识方法所检查的输入消息部分。各输入消息型式被记录于源表中，连同发现于界定在位置表的部分中的此型式的消息的任意数据。各源消息的各目标消息被记录于目标表中。各细部转换被记录于翻译细部表中。

对于各转换，有一被调用的方法（亦即执行所需的特殊转换的子程序），它由回到中间数据库中的方法表的引用所代表。方法表本身包含可执行的子程序（亦即一方法）的名称。当控制程序读取数据库时，此名称被控制程序所使用以导出适当的子程序的可执行的函数引用。各转换的最后结果回到此引用。然后决定引用以找出并执行子程序。而必须通过子程序的其它信息（参数信息）含有：一指针，源字段是在其暂时的计算机存储器中；一指针，目标字段是在其暂时的计算机存储器中；这些字段的尺寸（字节）；及一以查出将使用于非演算法转换的表（如果需要的话）。换句话说，分析的处理以形状与结构植入于需要被翻译的

数据库中，控制程序解译用于执行翻译的指令。

控制程序存在于正接收刺激的世界。实际上，其正给定输入数据而操作。

目前为止，控制程序被写入以从中间数据库取出数据命令需求。控制程序读取中间数据库，并使用所发现的指令以操纵其所给定的数据结构。

实际执行操纵的是一方法。例如，其中一个指令可以是“拷贝进入的第三字段至出去的 第六字段”。控制程序以字节位址的方式来定位第三字段。控制程序亦查出元数据库中的第三字段的尺寸。由控制程序所执行的下一个步骤是定位目标消息中的第六字段及其尺寸（字段）。依据从中间数据库所接收的方向，由控制程序所执行的下一个步骤是执行一方法，其可以是例如“拷贝”。执行方法通过目标与源的位置和长度。此方法执行从源位置至目标位置的拷贝的动作。然而，须注意控制程序并没有执行拷贝。它只有调用某些东西（亦即一方法），其可执行的函数引用是导自从中间数据库中找到的字“拷贝”。以此为例，在其它型式的信息中，重覆完全相同的处理。在中间数据库中的指令的指引下，控制程序计算源与目标数据的尺寸和位置，并调用适当的方法，通过此信息。控制程序的动作只限于计算参数及调用方法。它并不“知道”是由那一个方法执行。换句话说，它是一个中性程序。

每次方法执行时，所执行的方法告诉控制程序其是否成功地完成操作步骤。控制程序检查此传回值以确定各方法为成功的，藉以检查源至目标的转换已成功地完成。当控制程序已执行其欲执行的全部方法时，其将输出结构送至接收器。然后控制程序等待下一个进入信号或刺激。

在此处理的较佳结构中，可辨识消息的型式，与接收消息的速率比较，给定的翻译器所接收的消息的型式改变较慢。换句话说，翻译程序可以在消息的结构被送出者改变的前，接收许多拷贝的给定结构消息。计算机进行检查保持于碟片档中的中间数据库的处理是非常耗时的。必须读取许多数据库档，其已知是非常复杂而困难的。每次接收刺激时，计算机永远需要执行相同的计算，一再地寻找何处为第一字段、何处为第四字段。最好的方式是建造一翻译程序，当控制程序首先开始处理时，执行位置与长度的全部计算，将全部方法的本文名称转成函数指针。然后控制程序储存RAM存储器的指定部分中的中间数据库部分的缩写。此形成先前所提的数据阵列。此储存存储器的唯一目的是加速程序的操作。发现此储存存储器可改善操作速度约一千倍。观念是与使用于读写碟片档的处理中的快取存储器类似，除了在快取写入中并不会发生数据的操作。在快取中，计算机

可写入并拷贝出重复的指令。在此情形中，计算机不仅读取存储器，而且其亦将信息转换成更直接有用的形式。当程序首先开始处理时，发生中间数据库的版本的产生。当控制程序接收一进入消息、刺激（并非真实的元数据库）时，其使用控制程序的内部存储器结构，标示为包含于中间数据库中的相同数据的缩写版本。

翻译方法被写成子程序，此子程序接收一组特别的参数。各方法执行一特殊型式的转换。例如，只有一个执行拷贝，一个执行将字符格式数字转换成整数。各方法执行简单数据型式的特殊型式的操作。简单数据型式的数目非常小。因此，提供一组完整的此种方法。

应用方法是与翻译方法不同。如前所述，翻译程序具有此规定，全部的翻译方法取完全相同的参数。并规定会限制控制程序只能诠释由方法所采取的步骤，并保持控制程序的中性。此中性对本发明是重要的，因为其能使整个程序可以被容易地修改，且在不需要改变控制程序下可以扩充。此使方法可以相加以允许整个程序执行新的功能而不需通知（修改）控制程序。

当以任何程序语言来执行子程序时，需要一正式结构。一部分的此种正式结构为通过至子程序的参数。调用程序需要通过与程序希望接收的调用完全相同的参数，否则结果将会是无法预期的。换句话说，控制程序执行一特殊翻译方法，而翻译方法的参数是非常简单且永远是相同的。在翻译程序的情形中，这不会产生问题，由于全部可想到的翻译方法将需要相同参数数据。然后可以将新的翻译方法相加以顺利执行新的工作。这是因为参数是相同的。

在一般用途程序的情形中，无法预期一新方法所需的参数为何。然而，控制程序可以接受具有自己的新参数的新方法，而不需改变控制程序的需求是很重要的。

此问题的解决方式是藉由将应用方法所需的全部参数放入一数据结构中。如此将规定放在应用方法上，此应用方法接收结构形式的全部参数。在应用方法是由其它系统所写入的码而得到的情形中，其可能不是正确的。然而，坐（s i t）是一“包裹”程序的简单事项，将结构中的数据转换成分开的参数。应用方法所需要的参数结构是一可由翻译程序制作的数据结构。

欲解决此问题，应用方法得到不同的需求。首先，应用方法以不同于翻译方法的方式被调用。并非将参数给至应用方法，而是这些参数被储存于数据库表中。计算机所能保持的信息亦可以保持于数据结构中。于是，在普通的计算机程序中，

一程序能以其分开的参数来调用一普通的子程序。在此情形中，应用程序被调用，咨询数据库以寻找一参数，此参数为一数据结构其包含方法所需要的全部参数。无论方法需要为何种参数。换句话说，与分开片段的数据比较，结构的数据没有受限制。应用方法各具有一规定，它们的参数是做成储存结构的形式。应用方法的唯一的限制是它们的参数如何被设定。

应用翻译程序的技术于应用方法，我们建造一应用程序。第一个步骤，决定有应用程序的需要。程序设计者列出应用程序将接收的全部刺激消息。如同在翻译器中，各刺激进入一源表。各操作应用方法被做成一列表，应用方法将执行所需的操作及任何命令规定。这些应用方法调用各被记录成目标表中的一列。换句话说，各为一消息。各目标为一特殊消息，此消息是由两个部分构成（参见图6）。消息的第一部分永远是应用方法的参考数据。第二个部分是方法需要的参数结构的参考数据。下一个步骤是执行来自刺激的目标消息。其是以相同于只有翻译器的方式来执行：藉由辨识那一部分的进入或源消息产生一参数，然后使用其中一个方法“只设定一值”。换句话说，欲在消息开始时产生方法函数名称，使用一方法从查表中设定一值。（这是因为我们不希望看到与方法函数名称有关的任何部分刺激消息）。到目前为止，所发生的是程序的作用完全像是一翻译程序。然后并非将消息送至一接收程序作为目标消息（此为翻译程序所执行的），而是一新片段的控制程序（Server Shell）于内部使用。以相同于翻译程序分解其翻译方法的方式来分解方法引用。于是，将引用转换成函数指针，藉以调用此方法，将一指针引用通过至其它的消息作为一参数。然后控制程序检查传回码如同翻译程序以完成处理。

附录为包含应用写入程序的Sever Shells 与翻译器的设计文书的较佳实施例。图形为Gane 与Sarson 形式的流程图附有效述。

控制程序是一种程序可调用写成方法形式的基本计算步骤。其调用数据库以告诉或命令它在什么样的进入数据的环境下应使用何种方法。在控制程序的外产生基本步骤，在不需改变控制程序下可加入新的步骤。所以，对程序所做的改变亦可以改变方法或方法所使用的数据库。此处理几乎可消除重写程序的必要性。

说 明 书 附 图

图1

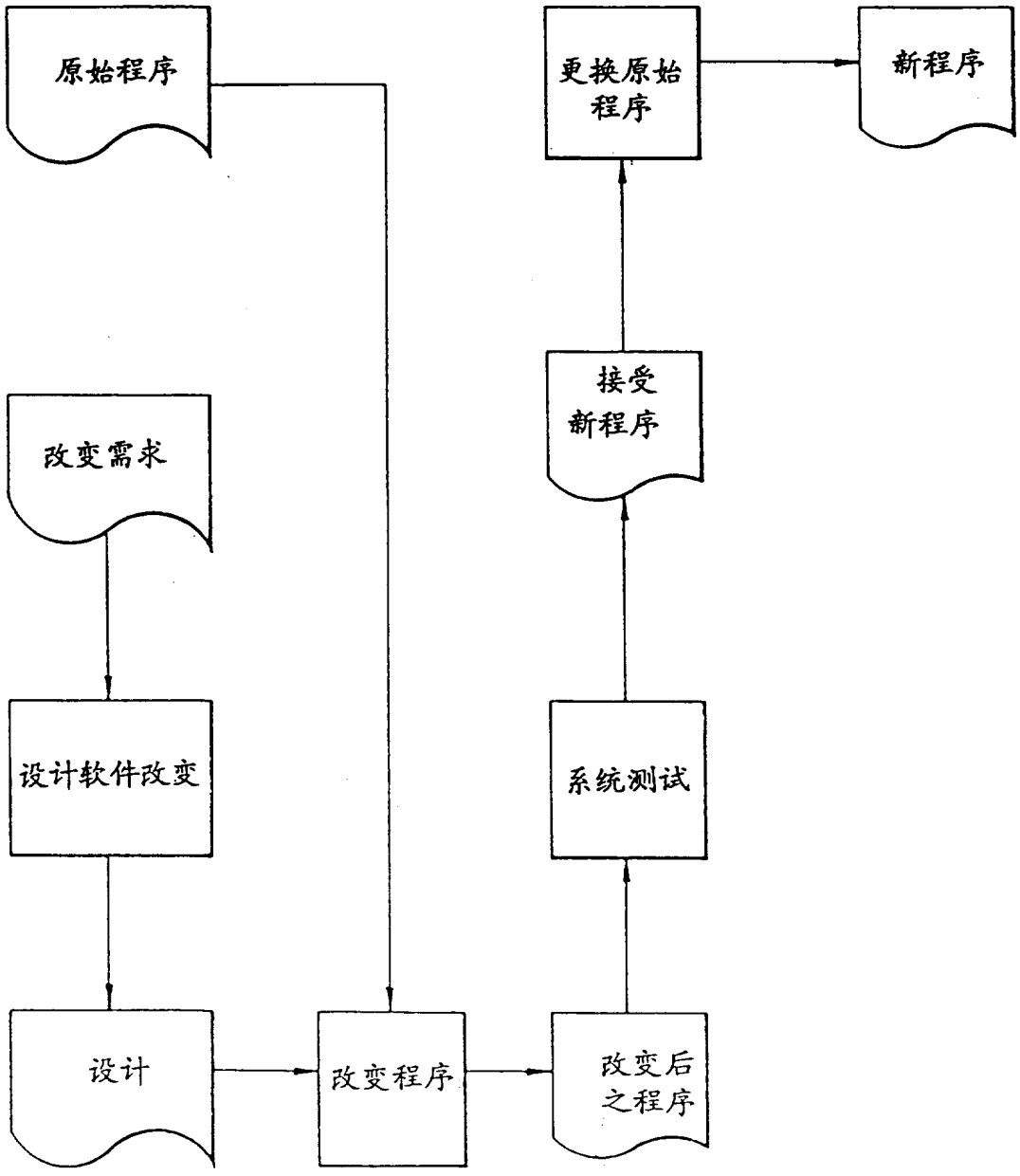


图2

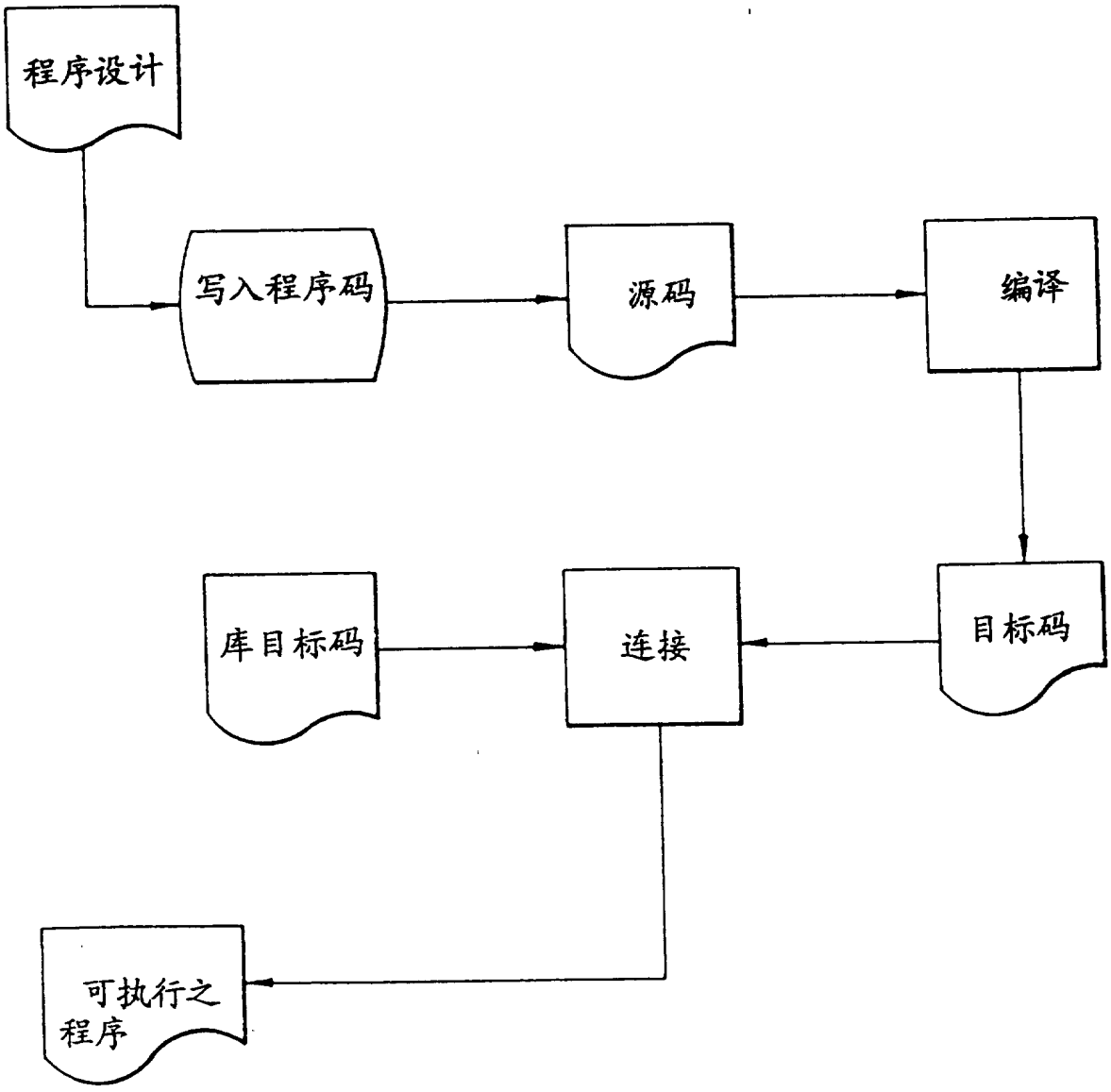


图3

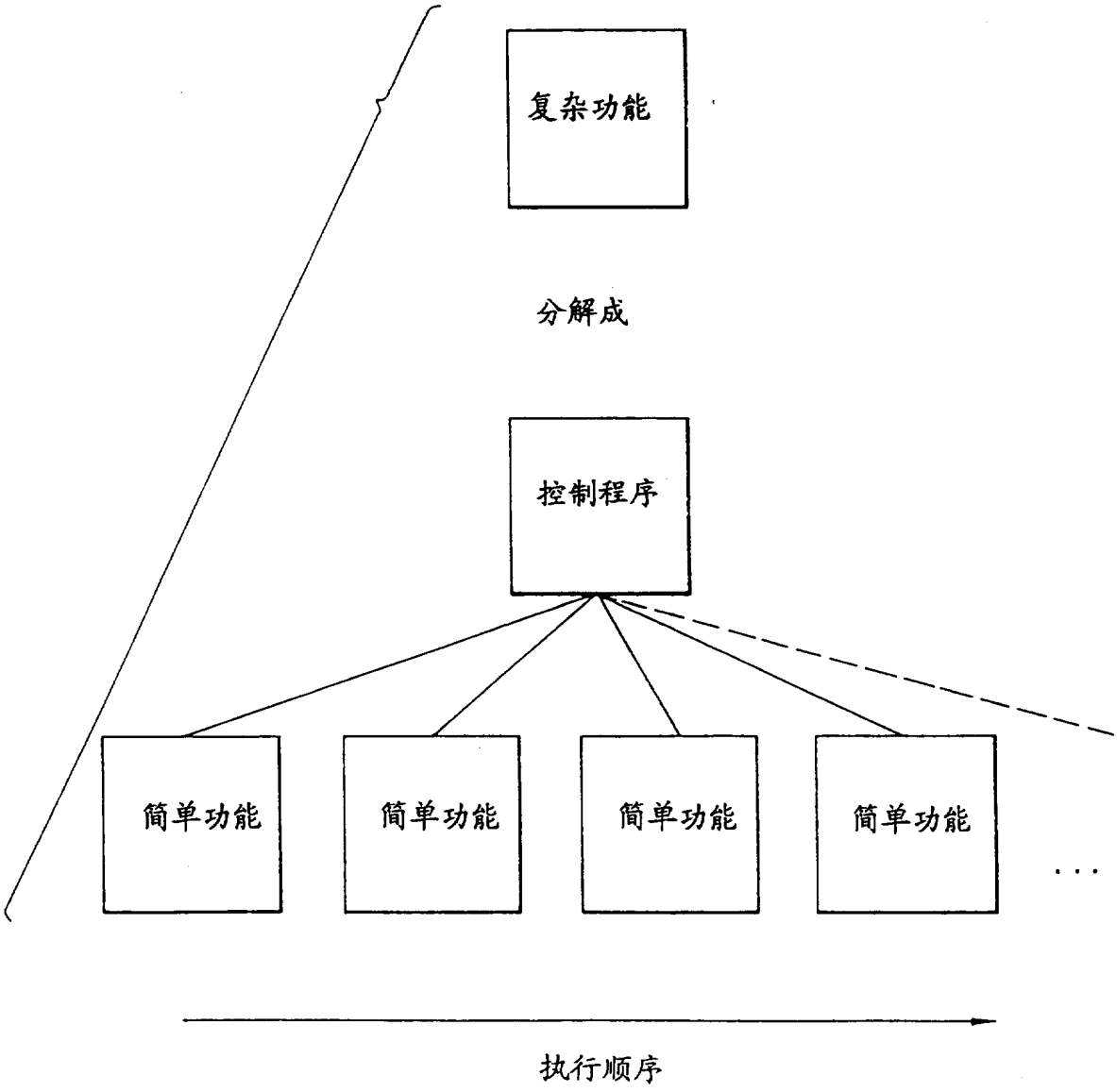


图4

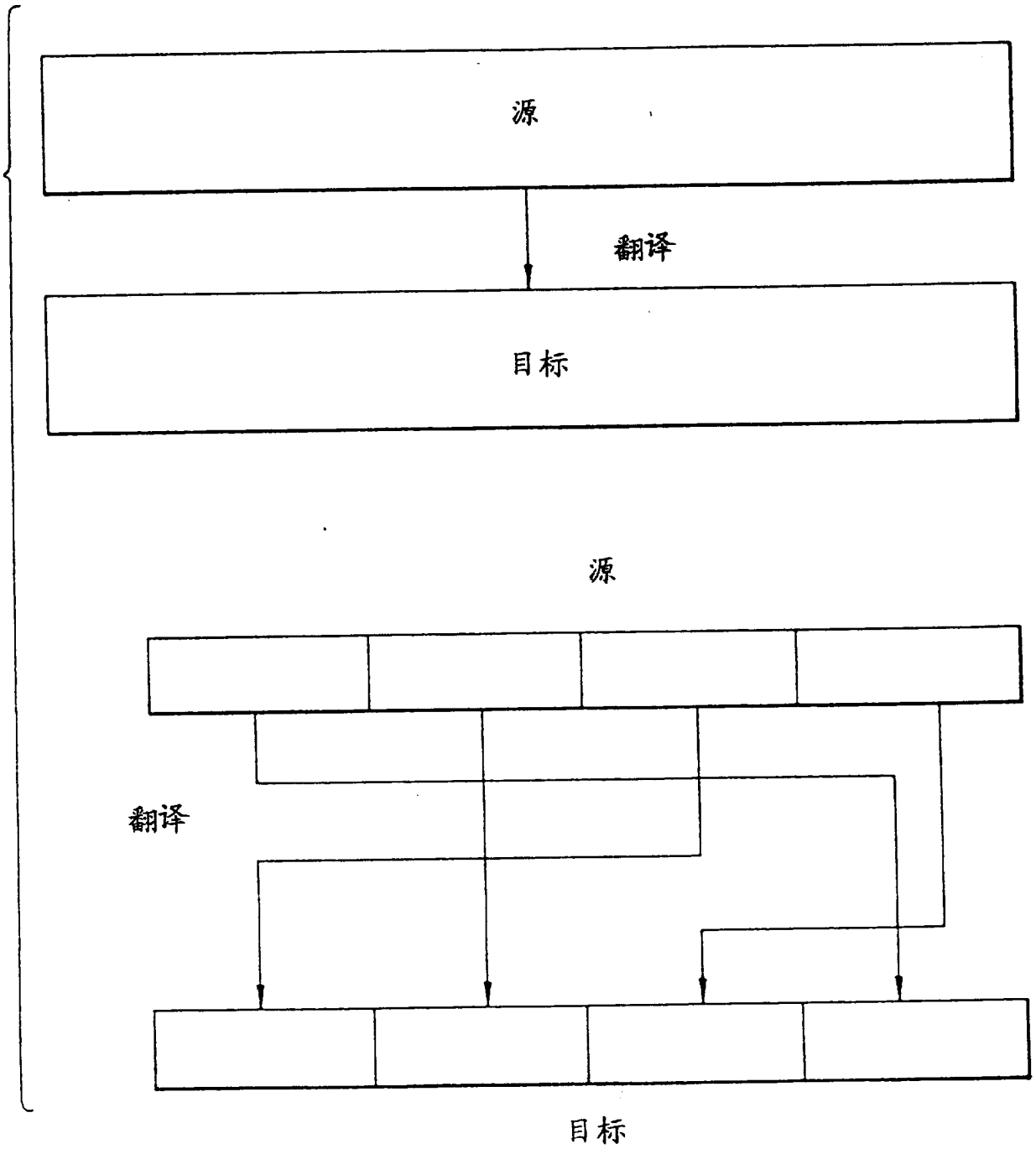


图5

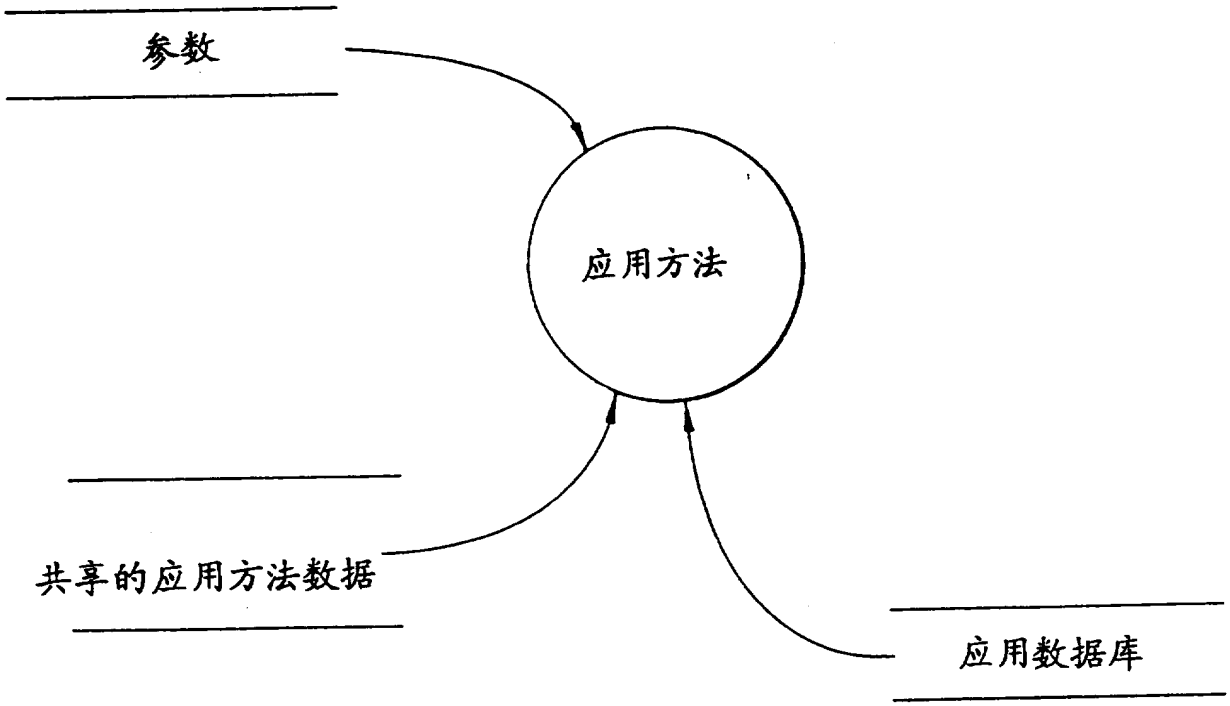


图6

方法引用	参数

图7

