[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 200610126321.1

[43] 公开日 2007 年 9 月 12 日

[11] 公开号 CN 101034348A

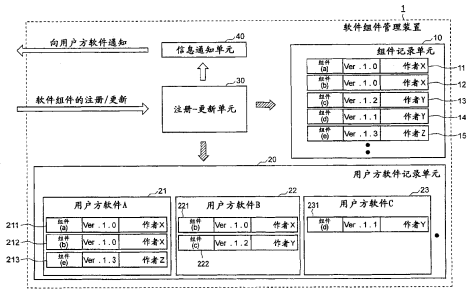
[22] 申请日 2006.8.31
[21] 申请号 200610126321.1
[30] 优先权
[32] 2006. 3. 8 [33] JP [31] 2006 - 062417
[71] 申请人 株式会社东芝
地址 日本东京都
[72] 发明人 高桥佳宏 池田信之 户谷浩隆
筱岛郁子 神胜雅
[74] 专利代理机构 中国国际贸易促进委员会专利商标事务所
代理人 康建忠
权利要求书 4 页 说明书 13 页 附图 9 页

[54] 发明名称

软件组件管理装置和方法

[57] 摘要

本发明提供了软件组件管理装置和方法。 组件记录单元记录多个软件组件，每个软件组件具有关键部分和补充部分。 用户方软件记录单元记录多个用户方软件，每个用户方软件与记录在组件记录单元中的至少一个软件组件相关联。 注册更新单元判定在所述组件记录单元中记录的软件组件的修改部分是关键部分还是补充部分，并且基于判定结果来更新所述组件记录单元和所述用户方软件记录单元中的至少一个的信息。 信息通知单元在修改部分是关键部分的情况下向与被修改的软件组件相关联的用户方软件通知该修改部分。



1. 一种管理软件组件的装置，包括:

组件记录单元，被配置来记录多个软件组件，每个软件组件具有关键部分和补充部分;

用户方软件记录单元，被配置来记录多个用户方软件，每个用户方软件与记录在所述组件记录单元中的至少一个软件组件相关联;

注册更新单元，被配置来判定在所述组件记录单元中记录的软件组件的修改部分是关键部分还是补充部分，并且基于判定结果来更新所述组件记录单元和所述用户方软件记录单元中的至少一个的信息;以及

信息通知单元，被配置来在修改部分是关键部分的情况下，向与被修改的软件组件相关联的用户方软件通知该修改部分。

2. 如权利要求 1 所述的装置，其中，

如果修改部分是关键部分并且修改部分的操作人员是被修改的软件组件的作者，

那么所述注册更新单元就更新在所述组件记录单元中被修改的软件组件的信息，以及

如果修改部分是关键部分并且修改部分的操作人员不是被修改的软件组件的作者，

那么所述注册更新单元就询问作者该软件组件的修改是否被许可。

3. 如权利要求 2 所述的装置，其中，

如果作者许可该软件组件的修改，

那么所述注册更新单元更新在所述组件记录单元中的被修改的软件组件的信息。

4. 如权利要求 2 所述的装置，其中，

如果作者不许可该软件组件的修改，

那么所述注册更新单元在所述组件记录单元中将被修改的软件组件作为新的软件组件注册。

5. 如权利要求 4 所述的装置，其中，

当新的软件组件被除了在所述用户方软件记录单元中记录的用户方软件以外的另一个用户方软件使用时，

那么所述注册更新单元将该另一个用户方软件作为与该新软件组件相关的新用户方软件注册在所述用户方软件记录单元中。

6. 如权利要求 1 所述的装置，其中，

如果该修改部分是补充部分，

那么所述注册更新单元在所述用户方软件记录单元中更新与修改软件组件的用户方软件相关的、被修改的软件组件的信息。

7. 如权利要求 6 所述的装置，其中，

当该被修改的软件组件被除了在所述用户方软件记录单元中记录的用户方软件以外的另一个用户方软件使用时，

那么所述注册更新单元将该另一个用户方软件作为与被修改的软件组件相关的新用户方软件注册在所述用户方软件记录单元中。

8. 如权利要求 1 所述的装置，其中，

软件组件被描述为源代码，并且

源代码具有区分关键部分和补充部分的标记信息。

9. 如权利要求 8 所述的装置，其中，

所述组件记录单元记录每个软件组件的源代码、名称、版本号和作者名称，以及

所述用户方软件记录单元记录与使用软件组件的用户方软件相关联的软件组件的名称、版本号，和作者名称。

10. 一种管理软件组件的方法，包括:

记录多个软件组件，每个软件组件具有关键部分和补充部分;

记录多个用户方软件，每个用户方软件与被记录的至少一个软件组件相关联;

判定被记录的软件组件的修改部分是关键部分还是补充部分;

基于判定结果来更新多个软件组件和多个用户方软件中的至少一个; 以及，

如果修改部分是关键部分，就向与被修改的软件组件相关联的用户方软件通知该修改部分。

11. 如权利要求 10 所述的方法，还包括:

如果修改部分是关键部分并且修改部分的操作人员是被修改的软件组件的作者，

那么就更新多个软件组件中被修改的软件组件的信息，以及

如果修改部分是关键部分并且修改部分的操作人员不是被修改的软件组件的作者，

那么就询问作者该软件组件的修改是否被许可。

12. 如权利要求 11 所述的方法，还包括:

如果作者许可该软件组件的修改，

那么就更新多个软件组件中被修改的软件组件的信息。

13. 如权利要求 11 所述的方法，还包括:

如果作者不许可该软件组件的修改，

那么就将被修改的软件组件作为新的软件组件注册。

14. 如权利要求 13 所述的方法，还包括:

当新的软件组件被除了记录的用户方软件以外的另一个用户方软件使用时，

将该另一个用户方软件注册为与该新软件组件相关的新用户方软件。

15. 如权利要求 10 所述的方法，还包括:

如果该修改部分是补充部分，

那么就更新与修改软件组件的用户方软件相关的、被修改的软件组件的信息。

16. 如权利要求 15 所述的方法，还包括:

当该被修改的软件组件被除了记录的用户方软件以外的另一个用户方软件使用时，

将该另一个用户方软件注册为与被修改的软件组件相关的新用户方软件。

17. 如权利要求 10 所述的方法，还包括:

软件组件被描述为源代码，并且

源代码具有区分关键部分和补充部分的标记信息。

18. 如权利要求 17 所述的方法，还包括:

多个软件组件的信息包括每个软件组件的源代码、名称、版本号和作者名称，以及

多个用户方软件的信息包括与使用软件组件的用户方软件相关联的软件组件的名称、版本号，和作者名称。

软件组件管理装置和方法

技术领域

本发明涉及用于管理用户方软件使用的软件组件的注册和更新的软件组件管理装置和方法。

背景技术

最近，作为使用软件的系统的开发形式或应用风格，利用合适的单元划分的软件被看作组件(以下，该被划分的软件被称为软件组件)。通过将多个开发的软件组件组合，就有效地创建和应用了新的软件。

作为用于注册的系统，软件组件管理装置管理和提供这样的软件组件。

在软件组件的用户方，用户方接收软件组件管理装置提供的一个或多个软件组件。通过组合所提供的多个软件组件，或者通过将（最先是由用户方开发的）软件模块安装到软件组件中，构建用户方的新软件。

通常来讲，存在多个软件组件，也存在多个用户方软件。因此，在软件组件管理装置中，用于集中管理哪个用户方软件使用（安装）提供的软件组件的功能是必需的。

此外，在预定软件组件中发现缺陷（诸如错误）的情况下，或者在修改（升级或版本更新）预定软件组件的情况下，需要向使用该软件组件的另一个用户（另一个用户方软件）通知该错误或版本更新。

用于满足这样的要求的 JP-A No.2001-282519（Kokai）（以下称为参考文件 1）是已知的。在参考文件 1 的软件组件管理装置中，注册了软件组件管理信息、用户方软件信息以及将软件组件管理信息和用户方软件信息相关联的链接信息。检索所有使用该预定软件组件的用户方软件，并且错误信息或修改被通知到所有检索到的用户方软件。

如今，从软件组件管理装置提供的软件组件通常被表示为源代码。因此，在用户方，该原始软件组件可以被容易地修改（改变）为适合于用户方软件的新的软件组件。

该修改并不总是软件组件关键部分的修改，而可能是补充部分的修改，例如纠错处理或输入/输出数据的参数调整。

以这种方式，在每个用户方软件中，存在着许多不同于“基础”原始软件组件的、具有补充部分的“分支”软件组件。

在背景技术中，用户方软件经常管理“分支”软件组件（管理修改历史）。换句话说，原始软件组件的供应源方并不集中管理所提供的软件组件的修改历史。

在这种情况下，即使用户方作出的软件组件的修改是对软件组件关键部分的修改（例如，关键部分中的功能扩展或缺陷消除），除了原始软件组件的作者之外的用户不能理解该变化给另一个用户方软件带来的影响的结果或程度。结果，很难恰当地向另一个用户方软件通知该修改。

另一个方面，如果某个用户方软件作出的软件组件的修改被通知到所有其他用户方软件，而不区分关键部分和补充部分，那么（接收到修改通知的）另一个用户方软件会执行软件组件的不必要的修改。结果，这样的修改通知甚至会造成坏影响。

发明内容

本发明是关于一种软件组件管理装置和方法，用于通过区分软件组件的关键部分和补充部分来有效地管理软件组件的修改。

根据本发明的一个方面，提供一种管理软件组件的装置，包括：组件记录单元，被配置来记录多个软件组件，每个软件组件具有关键部分和补充部分；用户方软件记录单元，被配置来记录多个用户方软件，每个用户方软件与记录在所述用户方软件中的至少一个软件组件相关联；注册更新单元，被配置来判定在所述组件记录单元中记录的软件组件的修改部分是关键部分还是补充部分，并且基于判定结果来

更新所述组件记录单元和所述用户方软件记录单元中的至少一个的信息；以及信息通知单元，被配置来在修改部分是关键部分的情况下向与被修改的软件组件相关联的用户方软件通知该修改部分。

根据本发明的另一个方面，提供一种管理软件组件的方法，包括：记录多个软件组件，每个软件组件具有关键部分和补充部分；记录多个用户方软件，每个用户方软件与被记录的至少一个软件组件相关联；判定被记录的软件组件的修改部分是关键部分还是补充部分；基于判定结果来更新多个组件和多个用户方软件中的至少一个；以及，如果修改部分是关键部分，就向与被修改的软件组件相关联的用户方软件通知该修改部分。

附图说明

图 1 是软件组件管理装置的应用概念。

图 2 是软件组件管理装置的系统结构。

图 3 是软件组件的源代码的一个例子。

图 4 是软件组件管理方法的处理的流程图。

图 5 是在其补充部分被改变的软件组件的更新情况下第一操作的示意图。

图 6 是在其补充部分被改变的软件组件的更新情况下第二操作的示意图。

图 7 是在其关键部分被改变的软件组件的更新情况下第一操作的示意图。

图 8 是在其关键部分被改变的软件组件的更新情况下第二操作的示意图。

图 9 是在其关键部分被改变的软件组件的更新情况下第三操作的示意图。

具体实施方式

以下，将参考附图描述本发明的各个实施例。本发明并不限于以

下实施例。

软件组件管理装置的结构

图 1 是一个实施例的软件组件管理装置 1 的应用概念的一个例子。

在软件组件管理装置 1 中，注册了用户创建的软件组件以及使用该软件组件的系统（下文中，该系统被称为用户方软件）。此外，该注册的软件组件被提供给另一个用户（另一个用户方软件）以便有效地利用软件资源。

例如，软件组件管理装置 1 是一种服务器装置。作为硬件结构，其配备了具有 CPU 的控制装置，具有 HDD 的存储装置，和用于经由电子通信电路（诸如 LAN）或存储介质发送/接收数据的输入/输出装置。

在本发明的软件组件管理装置 1 中，执行最近创建的软件组件的注册以及软件组件的供应的管理。此外，在修改软件组件（版本更新）的情况下，执行被修改的软件组件的更新注册（下文中，称为“更新”）。软件组件的修改历史也被管理。

软件组件的修改不仅仅由（原始）软件组件的作者（在图 1 中，用户方软件 A 的维护者）执行，而且在预定条件下也由除了作者外的其他用户（图 1 中，用户方软件 B、C 和 D 的维护者）执行。

如下所述，在修改（改变）软件组件关键部分的情况下，关于软件组件的关键部分已经被修改的通知被发送到所有使用该软件组件的用户方软件。在这种情况下，当用户方软件 B、C 和 D 的用户（维护者）修改该软件组件的关键部分时，对于许可修改软件组件的请求被发送到软件组件的作者。如果从作者接收到修改软件组件的许可，那么就更新（修改）该软件组件。

在图 1 中，用户方软件 A 的维护者是软件组件的作者。在将软件组件注册到软件组件管理装置 1 之后，用户方软件 B、C 和 D 的其他维护者（用户）利用该软件组件。然而，访问该软件组件管理装置 1

的用户方软件的数量并不限于图 1 的 4 个。此外，可以从用户方软件 B、C 和 D 执行新软件组件的创建/注册。

图 2 是软件组件管理装置 1 的系统结构的框图。该软件组件管理装置 1 包括组件记录单元 10，用户方软件记录单元 20，注册更新单元 30 和信息通知单元 40。

在组件记录单元 10 中，记录软件组件的源代码（参考图 3），软件组件的管理信息（诸如组件名称、版本号，和作者名称）。

在图 2 中，作者 X 创建的组件（a）11 和组件（b）12 被记录为版本号 Ver.1.0。作者 Y 创建的组件（c）13 被记录为版本号 Ver.1.2，作者 Y 创建的组件（d）14 被记录为版本号 Ver.1.1，而作者 Z 创建的组件（e）15 被记录为版本号 Ver.1.3。在该情况下，除了 Ver.1.0 之外的版本号表示该组件已经被修改了至少一次。

在用户方软件记录单元 20 中，利用记录在组件记录单元 10 中的用户方软件与该软件组件相关地被记录。

例如，在图 2 中，用户方软件 A 利用记录在组件记录单元 10 中的软件组件（a）（b）（c）。在用户方软件 A 的记录列 21 中，记录这些软件组件的管理信息（（a）211，（b）212，（e）213）。

以相同的方法，用户方软件 B 利用记录在组件记录单元 10 中的软件组件（b）（c）。在用户方软件 B 的记录列 22 中，记录这些软件组件的管理信息（（b）221，（c）222）。

另外，以相同的方法，用户方软件 C 利用记录在组件记录单元 10 中的软件组件（d）。在用户方软件 C 的记录列 23 中，记录这些软件组件的管理信息（（d）231）。

即使多个软件组件的每个组件名称相同（和原始软件组件的名称相同），不同版本的每个软件组件也经常用于不同的用户方软件中。因此，在本实施例中，在每个用户方软件中的软件组件的修改历史由版本号管理。为了清楚地解释这个方面，在图 2 中，对于具有相同名称的多个软件组件，对于每个用户方软件向每个软件组件分配不同的符号（号码）（例如，对于组件（b），分别分配了 212 和 221）。此

外，在组件记录单元 10 和用户方软件记录单元 20 中，对相同名称的两个软件组件分配不同的符号（例如，对于组件（a），分别分配了 11 和 211）。

在注册更新单元 30 中，分析用户（软件组件的作者和更新者）输入的软件组件的源代码和管理信息，并且在组件记录单元 10 和用户方软件记录单元 20 中的至少一个中注册/更新该软件组件。以下将解释注册更新单元 30 的操作。

在信息通知单元 40 中，各种信息被通知到记录在组件记录单元 10 的用户方软件的每个维护者（用户）。特别地，当注册更新单元 30 判定修改的软件组件的修改（改变）部分是关键部分时，该修改信息就被通知到每个维护者（每个用户方软件）。

图 3 是软件组件管理装置 1 处理的软件组件的源代码的一个例子的示意图。

通常，软件组件的源代码具有明显不同的关键部分和补充部分。关键部分是软件组件实现的功能的核心部分。补充部分不是核心部分，例如为执行纠错处理的部分、创建操作参数的部分、检查输入/输出限制的部分以及控制用户接口的部分。

在图 3 的源代码中，为了使关键部分清楚，在关键部分开始位置之前插入了注释语句（/*<essential part>*/），并且在关键部分结尾位置之后插入了注释语句（/*<essential part>*/）。以这种方法，为了使补充部分清楚，在补充部分开始位置之前插入了注释语句（/*<supplemental part>*/），并且在补充部分结尾位置之后插入了注释语句（/*<supplemental part>*/）。

一般而言，在多个用户方软件使用一个软件组件的情况下，多个用户方软件中共同使用该关键部分。补充部分通常被修改成适合于每个用户方软件使用的形式并被每个用户方软件不同地使用。

在修改（改变）关键部分的情况下，该修改影响了所有的用户方软件，并且修改带来的影响程度非常高。另一方面，在修改补充部分的情况下，该修改通常仅仅影响一些用户方软件，并且该修改带来的

影响程度相对较低。

例如，在预定用户方软件中，加到关键部分的用于功能扩展和错误消除的修改通常对于其他用户方软件是必需的，以便作出共同的反映。另一方面，在预定用户方软件中，加到补充部分的修改通常是预定用户方软件特有的，而对其他用户方软件来说不是必需的。

如图 3 所示，在本发明的软件组件管理装置 1 中，注册具有源代码的软件组件，该源代码的关键部分和补充部分被区别地描述。

最有效地是，软件组件的作者通过区分关键部分和补充部分来描述源代码，这是因为作者知道原始软件组件的内容。但是，描述该源代码的操作者并不限于该作者。

此外，区别地描述关键部分和补充部分的具体方法并不限于图 3 中的利用注释语句的方法。简而言之，软件组件管理装置 1 可以机械地（自动地）读取关键部分和补充部分，并且源代码的说明并不影响该软件组件的运行。在该条件下，可以采用各种描述该源代码的方法。

（2）管理软件组件的方法

接下来，解释（如上所述构造的）软件组件管理装置 1 的操作。图 4 是软件组件管理方法的处理的流程图。在该流程图中的该处理由软件组件管理装置 1 中的注册更新单元 30（参考图 2）执行。

首先，在 ST0，输入与软件组件管理装置 1 的用户（软件组件的作者或更新者）创建的软件组件相关的数据。例如，该数据包括软件组件的名称、利用该软件组件的用户方软件的名称，软件组件的源代码和作者（或更新者）的名称。

数据的输入形式没有特别限制。例如，可以采用经由电子通信电路（诸如 LAN，因特网或专用线）的输入形式。替换地，可以采用经由存储介质（诸如光盘、磁盘或半导体存储器）的输入形式。

接下来，在 ST1 中，判定输入软件组件是在组件记录单元 10 和用户方软件记录单元 20 中要被更新的软件组件，还是要在组件记录单元 10 和用户方软件记录单元 20 中注册的新软件组件。该判定通过检

查记录的每个软件组件的名称来执行的。

在新软件组件的情况下，处理前进到新注册（ST13）。另一方面，在更新记录的软件组件的情况下，处理前进到更新（ST2）。

在 ST2 中，输入软件组件的源代码和具有与输入软件组件相同的名称的注册软件组件的源代码相比较。然后判定输入软件组件的修改部分是关键部分还是补充部分。

如图 3 所示，源代码区别地具有关键部分和补充部分。通过将输入软件组件的源代码和具有与输入软件组件相同的名称的注册软件组件的源代码相比较，就检测到了输入软件组件和注册软件组件之间的不同部分。从而，判定出输入软件组件的修改部分（改变部分）是关键部分还是补充部分。

在软件组件的修改部分是补充部分（在 ST3 为“否”）时，处理前进到 ST7。

在 ST7 中，判定使用修改软件组件的用户方软件是已经在用户方软件记录单元 20 注册还是新用户方软件。

在用户方软件已经注册在用户方软件记录单元 20 中的情况下(在 ST7 为“否”)，更新与注册用户方软件相关的修改的软件组件的管理信息（ST15）。

另一方面，在用户方软件是新用户方软件的情况下（在 ST7“是”），在用户方软件记录单元 20 中创建新用户方软件的记录列（ST8）。该新用户方软件被注册在用户方软件记录单元 20 的记录列中。

图 5 显示了解释在更新其修改部分是补充部分的软件组件的情况下操作的例子。

在该例子中，用户方软件 B 的操作人员（更新者）V 修改用户方软件 B 使用的软件组件中的组件（b）。具体地，仅仅修改了组件（b）的补充部分，并且组件（b）的版本号从 Ver.1.0 更新到 Ver.1.1。

因为修改部分是补充部分，因此注册更新单元 30 的处理从 ST3 前进到 ST7。此外，处理从 ST7 前进到 ST15，因为用户方软件 B 已

经在用户方软件记录单元 20 中注册了。在用户方软件 B 的记录列 22 中的组件（b）的管理信息 221a 中，版本号从 Ver.1.0 更新到 Ver.1.1。

以这种方法，在更新用户方软件记录单元 20 中的软件组件的情况下，仅仅更新在此时修改软件组件的用户方软件 B 的管理信息，而不更新另一个用户方软件的管理信息。例如，用户方软件 A 使用组件（b）。但是，在用户方软件 A 中，组件（b）没有被修改（改变）并且其被用作为 Ver.1.0。因此，用户方软件 A 的组件（b）212 的管理信息没有被更新。

此外，在组件（b）的修改（改变）部分是补充部分的情况下，组件记录单元 10 中的组件（b）的管理信息不被更新。如后要解释的，组件记录单元 10 中的组件的管理信息仅仅在组件的关键部分被修改（改变）的情况下被更新。

在组件记录单元 10 中，作为基础的软件组件（原始软件组件）的管理信息被作为一个组件存储。另一方面，在用户方软件记录单元 20 中，作为分支的软件组件（修改的软件组件）对应于每个用户方软件而被存储。该分支指的是由用户方软件产生的基础的修改版本。简而言之，组件记录单元 10 管理软件组件的基础，用户方软件记录单元 20 管理软件组件的分支。因此，两种管理被有效地共享。

此外，在软件组件的修改部分是补充部分的情况下，信息通知单元 40 不向其他用户方软件通知修改部分的信息。如上所述，补充部分的修改几乎是预定用户方软件特有的。如果补充部分的这种修改被通知给所有用户方软件，那么每个用户方软件通常接收了不是那么重要（并不总是与每个用户方软件相关）的通知，使得每个用户方软件感到混乱。结果，降低了用户方软件的研发阶段的效率，并且妨碍了在用户方软件的使用阶段的顺利应用。

因此，在修改部分是补充部分的情况下不通知到另一个用户方软件，相当大地提高了用户方软件的开发效率，并且继续了用户方软件的顺利应用。

图 6 显示了一个例子，用于解释在软件组件的修改部分是补充部

分并且修改软件组件的用户方软件没有被注册的情况下，注册更新单元 30 的操作。

改变组件(b)的补充部分以及将版本号从 Ver.1.0 更新到 Ver.1.1 的特征与图 5 的例子相同。但是，更新者 W 是新用户方软件 D 的操作人员的特征不同于图 5。

在该情况中，注册更新单元 30 在用户方软件记录单元 20 中创建新用户方软件的记录列（用户方软件 D 24）（图 4 中的 ST8）。之后，组件（b）的管理信息作为版本号 Ver.1.1 注册在用户方软件 D 中。

接下来，解释软件组件的修改部分是关键部分的情况下的更新操作。在修改部分是关键部分的情况中（图 4 的 ST3），更新软件组件的操作人员（用户）被确定为软件组件的原始版本的作者（ST4）。

通常，原始版本的作者可以具有软件组件内容的最多知识。如果操作人员是作者，那么许可软件组件的更新。在这种情况下，修改部分是关键部分。从而，在组件记录单元 10 中的软件组件（基础）的源代码和管理信息被更新（ST5）。

此外，软件组件的关键部分的修改经常影响其他用户方软件（该软件组件的相同修改应当在另一个用户方软件中运行）。因此，信息通知单元 40 将关键部分的修改的信息通知给所有利用该软件组件的用户方软件（ST6）。

在关键部分修改的情况和补充部分修改的情况中，从图 4 的 ST7 开始的处理是相同的。因此，省略了对其的解释。

图 7 显示了解释关键部分改变的情况下注册更新单元 30 的上述操作的例子。具体而言，修改（改变）组件（b）的关键部分，并且操作人员（更新者）是组件（b）的原始版本的作者 A。

在该情况下，注册更新单元 30 更新（修改）在组件记录单元 10 中注册的组件（b）的源代码，并将组件（b）的管理信息的版本号从 Ver.1.0 更新到 Ver.1.1。

此外，对于所有利用组件（b）的维护者（用户方软件），信息通知单元 40 通知有关组件（b）的关键部分被更新的信息。在该通知

中，可以添加更新的源代码。

例如，软件组件的关键部分被改变来延伸软件组件的功能或者修补软件组件的错误。该修改可能对于所有使用该软件组件的用户方软件来说是重要的（必须的）。响应于该通知，维护者（用户方软件）可以通过向更新者询问或分析附带的源代码很快地理解改变的内容。从而提高了用户方软件的开发效率或使用效率。

仅仅对于更新者的用户方软件（在图 7 中，是用户方软件 A）运行在用户方软件记录单元 20 中的管理信息的更新。对于其他用户方软件，每个用户方软件决定采用该改变。在从另一个用户方软件接收到更新请求之后，在用户方软件记录单元 20 中的该另一个用户方软件的管理信息被更新。

如果软件组件的改变部分是关键部分并且更新者不是软件组件的原始版本的作者（ST4 中的“否”），那么许可更新软件组件的请求被发送到作者（ST10）。例如，信息通知单元 40 可以询问作者的用户方软件或者可以通过电子邮件询问作者。

响应于该询问，作者判定软件组件的更新是否被许可，并经由信息通知单元 40 或电子邮件答复判决结果。

注册更新单元 30 判定是否从作者接收了更新许可（ST11）。在接收更新许可的情况下，以与更新者是作者的情况下相同的方式执行软件组件的更新处理（ST5 之后的步骤）。

图 8 显示了解释在关键部分的更新请求不是由作者 X 而是由用户方软件 B 的更新者 V 进行的情况下上述操作的例子。

注册更新单元 30 向作者 X 询问组件（b）的更新许可。在从作者 A 接收到更新许可的情况下，注册更新单元 30 更新注册在组件记录单元 10 中的组件（b）。在该情况下，通过改变关键部分，组件（b）的实际作者可以从 X 改变为 V。在图 8 中，除了将版本号从 Ver.1.0 更新到 Ver.1.1，作者名称也从 X 更新到 V。

此外，注册更新单元 30 将有关组件（b）的关键部分被更新的信息经由信息通知单元 40 发送到组件（b）的所有维护者（用户方软件）。

在用户方软件记录单元 20 中更新管理信息的情况下，以与其他例子相同的方式，仅仅更新修改组件（b）的（实际上安装了被修改的组件（b））用户方软件的组件（b）的管理信息。

另一个方面，假设除了作者 X 以外的用户 V 修改了软件组件的关键部分，并且向作者 X 请求更新许可。如果没有获得来自作者 X 的更新许可（ST11 中的“否”），那么被修改软件组件被认为是用户 V 创建的新的软件组件。

简而言之，在用户方软件记录单元 20 中，删除了用户 V 的用户方软件的软件组件（没有获得来自作者的更新许可）的管理信息。在组件记录单元 10 中，其关键部分被修改的、作为新软件组件的软件组件的源代码以及管理信息被新注册（ST13）。

此外，在用户方软件记录单元 20 中，代替被删除的管理信息，新软件组件的管理信息被记录在用户 V 的用户方软件的栏中（ST14，ST9）。

图 9 是解释没有从作者获得更新许可的情况下操作的具体例子。在图 9 中，改变组件（b）的关键部分的更新者不是组件（b）的作者 X 而是用户方软件 B 的用户 V（维护者）。另外，没有从作者 X 获得组件（b）的更新许可。

在该情况中，在组件记录单元 10 中，（原始）组件（b）的源代码和管理信息 12 被保持为版本号 V.1.0，而不更新。

另一方面，关键部分被改变的组件（b）被作为新的软件组件（f）16 注册在组件记录单元 10 中。在该情况下，新软件组件（f）的版本号是 V.1.0，并且用户 V（更新者）作为组件（f）的作者被注册。

此外，在用户方软件记录单元 20 的用户方软件 B 中，组件（b）221 被删除，而新组件（f）223 被注册。

原始地，对于组件（b），所有部分（包括关键部分）没有被改变。因此，对于利用组件（b）的所有用户（用户方软件），不发送通知。

如上所述，在本发明的软件组件管理装置 1 中，组件（基础）的

源代码和管理信息被注册在组件记录单元 10 中。此外，软件组件（分支）的管理信息与（使用软件组件的）每个用户方软件相对应地被注册在用户方软件记录单元 20 中。因此，软件组件可以被有效地管理而不会遗漏。

此外，在改变注册的软件组件的情况下，判定该改变部分是软件组件的关键部分还是补充部分。如果改变部分是关键部分，那么关键部分的改变被通知给所有利用该软件组件的用户方软件（用户）。如果改变的部分是补充部分，那么就不会发送通知。因此，阻止了不必要信息的泛滥，并且只有必要的信息会准确地提供给其他用户。结果，在用户方软件的开发阶段提高了开发效率，并且在用户方软件的应用阶段促进了顺利的使用。

此外，当软件组件被除了该软件组件的原始版本的作者之外的用户修改时，只有获得了来自作者的更新许可，该软件组件才被修改所更新。因此，可以预先防止错误的修改带来的软件组件的更新。

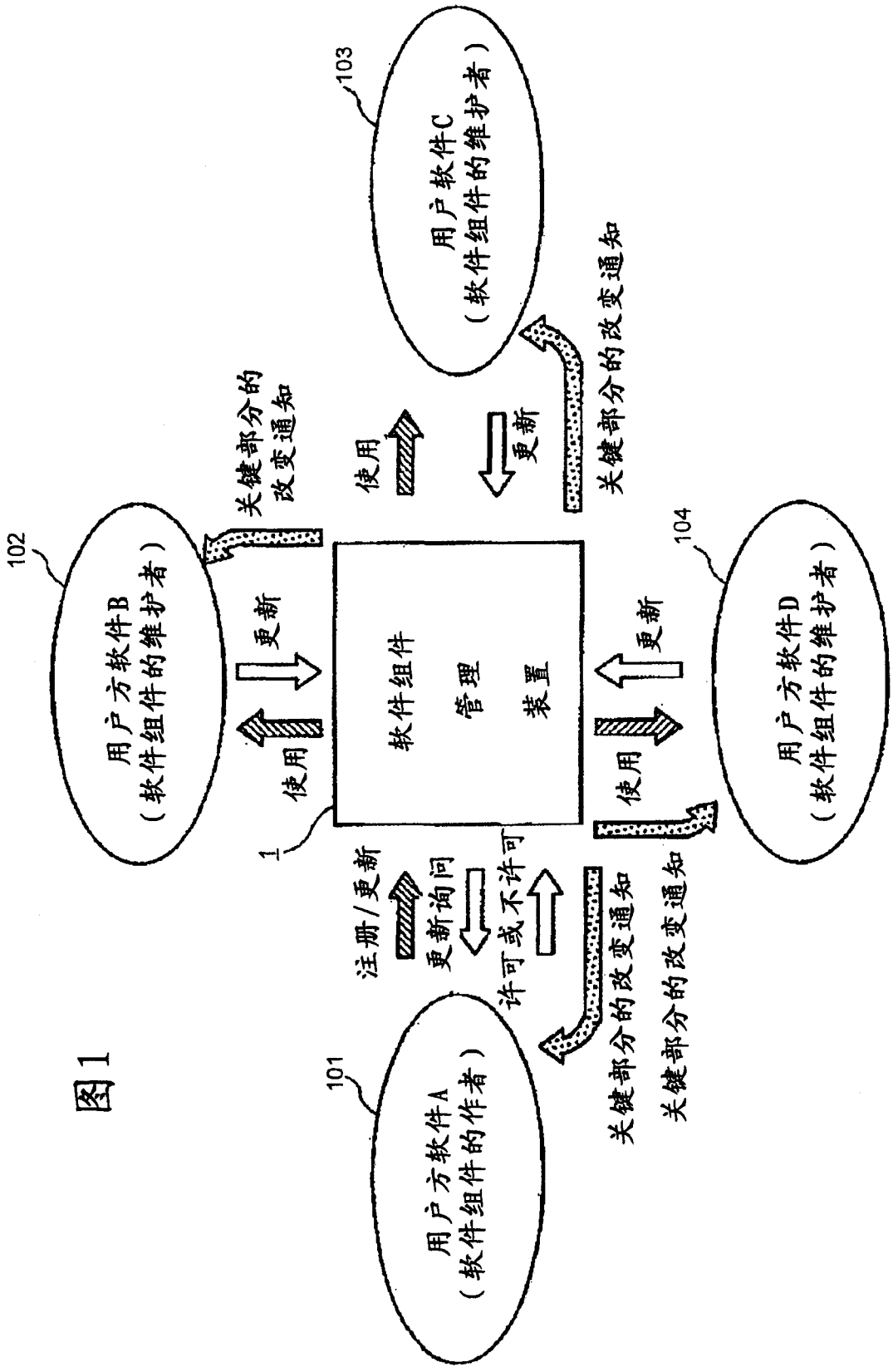

用户方软件A
（软件组件的作者）
101
用户方软件B
（软件组件的维护者）
102
用户方软件C
（软件组件的维护者）
103
用户方软件D
（软件组件的维护者）
104
1
软件组件
管理
装置
关键部分的
改变通知
使用
更新
注册/更新
更新询问
许可或不许可
关键部分的改变通知
关键部分的改变通知

图1

图2

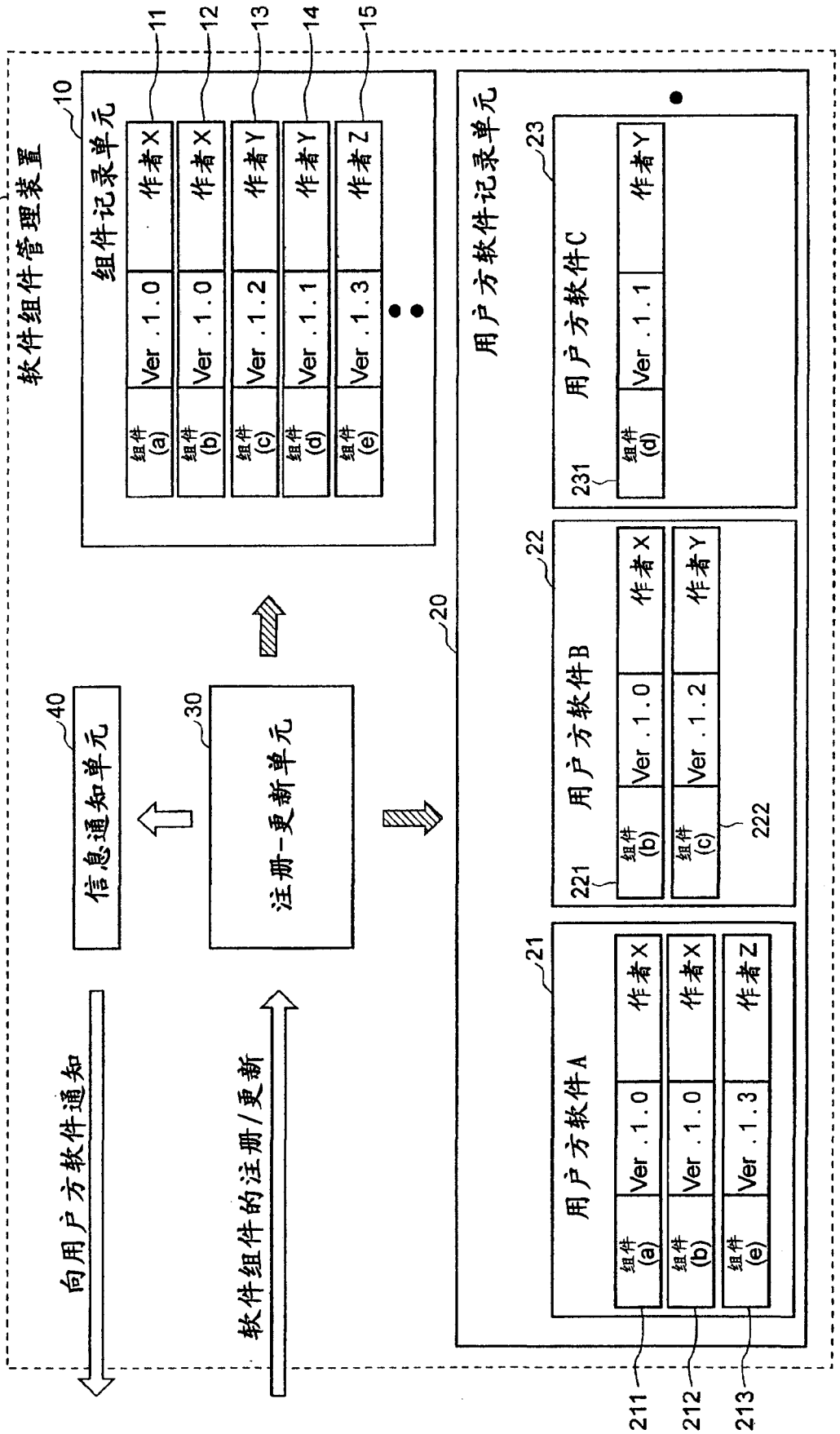

1
软件组件管理装置
组件记录单元
10
11
12
13
14
15
组件(a) Ver .1.0 作者X
组件(b) Ver .1.0 作者X
组件(c) Ver .1.2 作者Y
组件(d) Ver .1.1 作者Y
组件(e) Ver .1.3 作者Z
用户方软件记录单元
20
用户方软件A
21
组件(a) Ver .1.0 作者X
组件(b) Ver .1.0 作者X
组件(e) Ver .1.3 作者Z
用户方软件B
22
221
组件(b) Ver .1.0 作者X
组件(c) Ver .1.2 作者Y
222
用户方软件C
23
231
组件(d) Ver .1.1 作者Y
信息通知单元
40
注册-更新单元
30
向用户方软件通知
软件组件的注册/更新
211
212
213

图 3

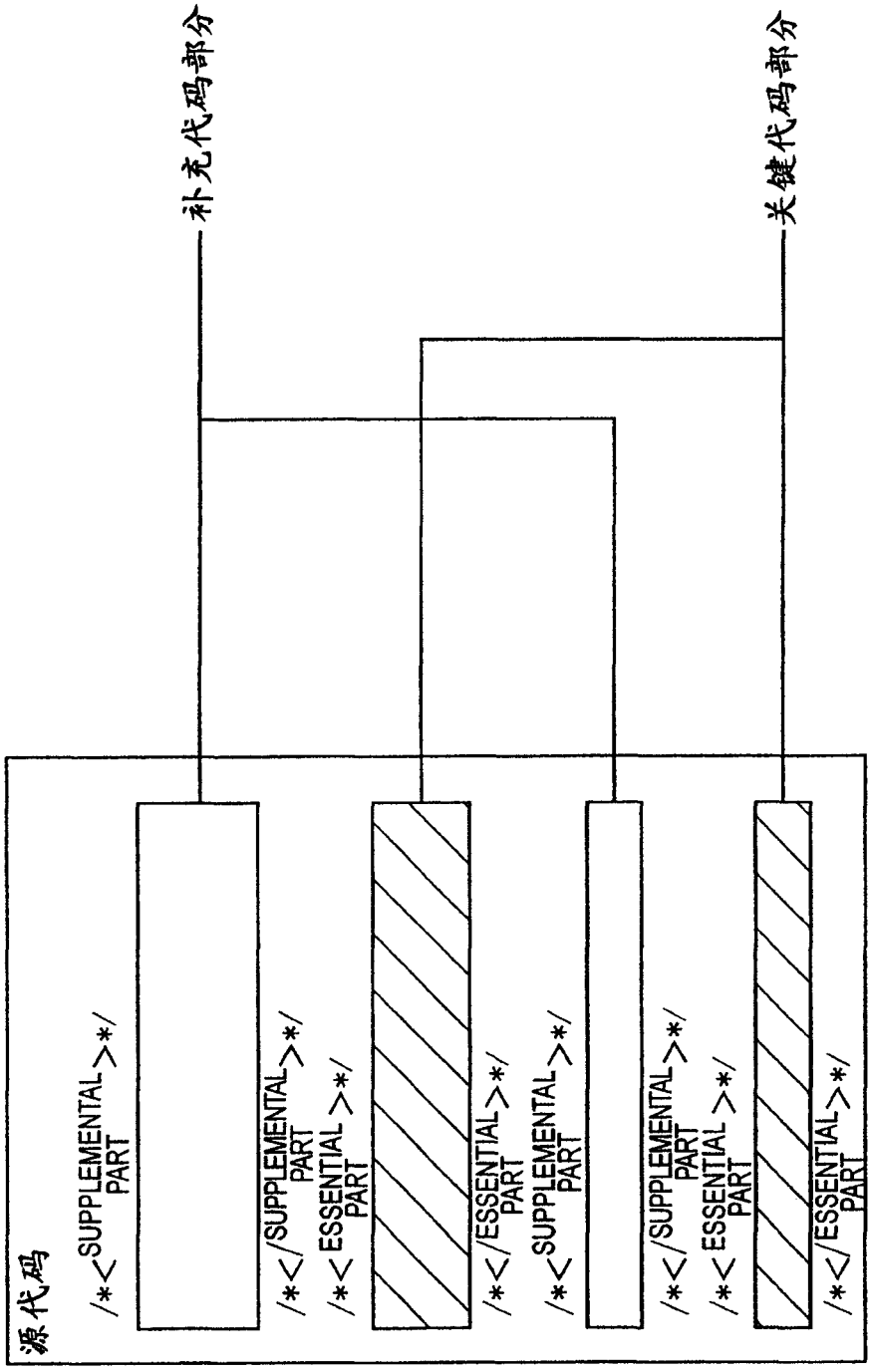

源代码
/*<SUPPLEMENTAL>*/
PART
/*</SUPPLEMENTAL>*/
PART
/*<ESSENTIAL>*/
PART
/*</ESSENTIAL>*/
PART
/*<SUPPLEMENTAL>*/
PART
/*</SUPPLEMENTAL>*/
PART
/*<ESSENTIAL>*/
PART
/*</ESSENTIAL>*/
PART
补充代码部分
关键代码部分

图4

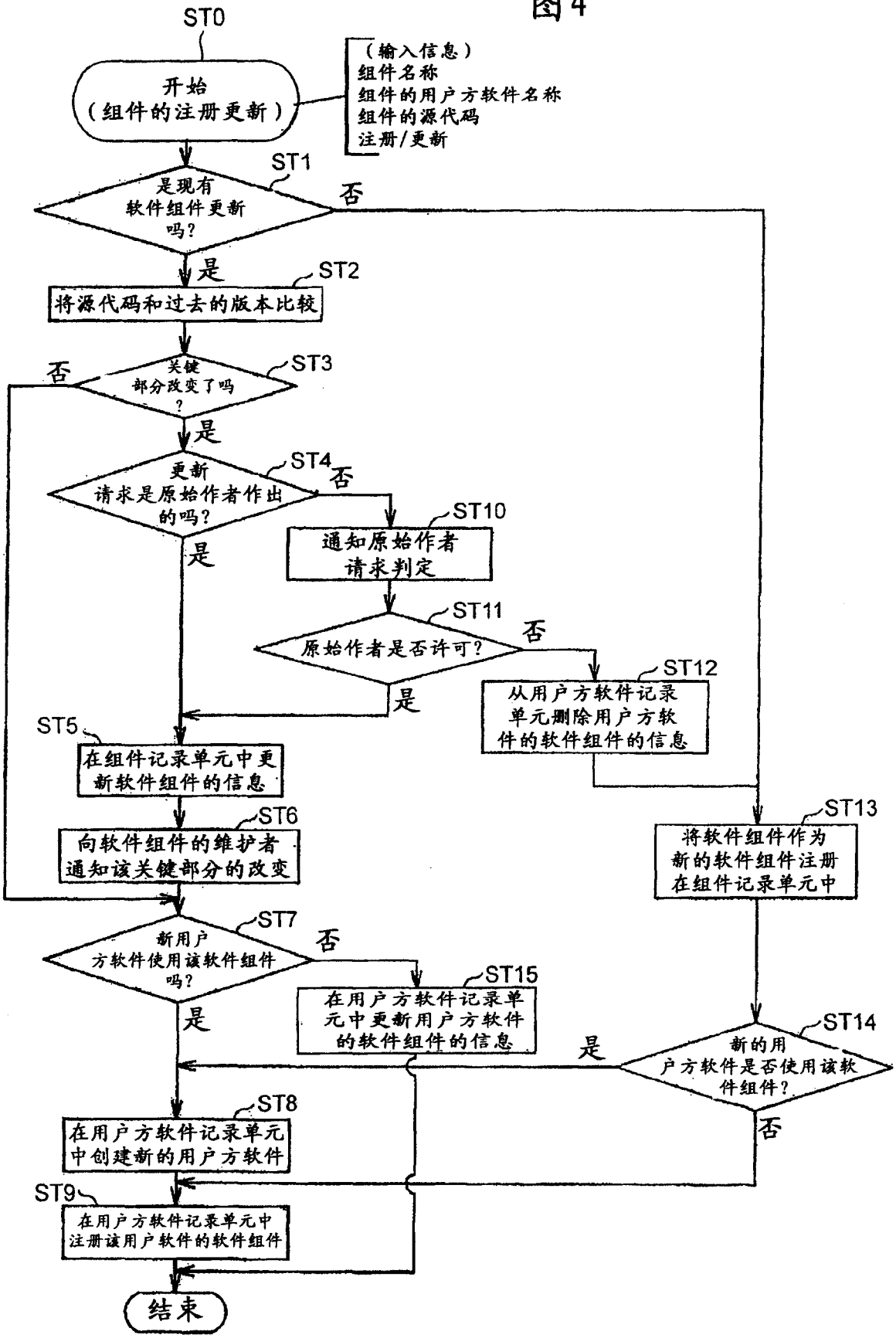

ST0
开始
（组件的注册更新）
（输入信息）
组件名称
组件的用户方软件名称
组件的源代码
注册/更新
ST1
是现有软件组件更新吗？
否
是
ST2
将源代码和过去的版本比较
ST3
关键部分改变了吗？
否
是
ST4
更新请求是原始作者作出的吗？
否
是
ST10
通知原始作者请求判定
ST11
原始作者是否许可？
否
是
ST12
从用户方软件记录单元删除用户方软件的软件组件的信息
ST5
在组件记录单元中更新软件组件的信息
ST6
向软件组件的维护者通知该关键部分的改变
ST13
将软件组件作为新的软件组件注册在组件记录单元中
ST7
新用户方软件使用该软件组件吗？
否
是
ST15
在用户方软件记录单元中更新用户方软件的软件组件的信息
ST14
新的用户方软件是否使用该软件组件？
是
否
ST8
在用户方软件记录单元中创建新的用户方软件
ST9
在用户方软件记录单元中注册该用户软件的软件组件
结束

图5

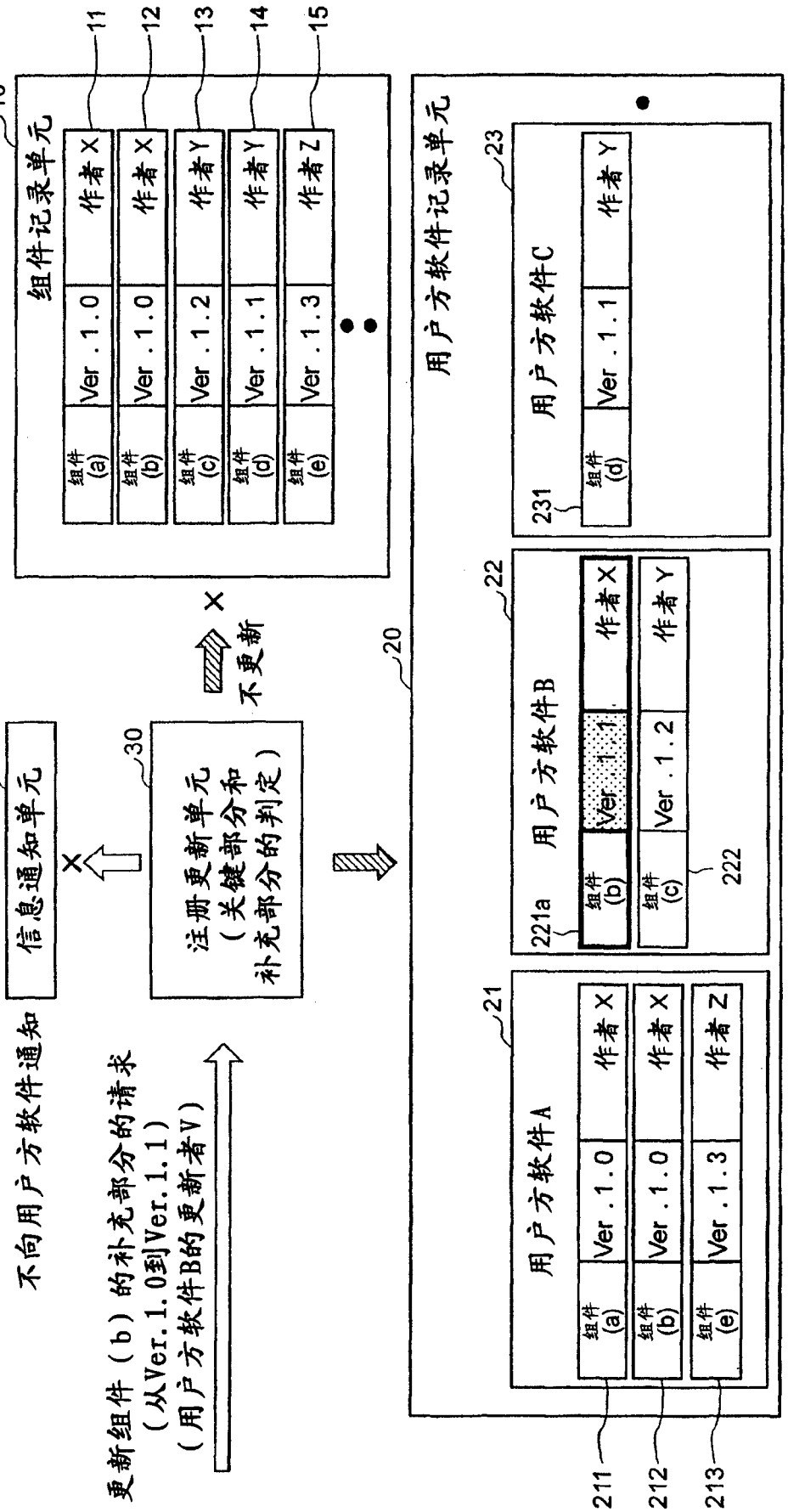

10
组件记录单元
11
12
13
14
15
组件(a) Ver .1.0 作者X
组件(b) Ver .1.0 作者X
组件(c) Ver .1.2 作者Y
组件(d) Ver .1.1 作者Y
组件(e) Ver .1.3 作者Z
20
用户方软件记录单元
21
用户方软件A
组件(a) Ver .1.0 作者X
组件(b) Ver .1.0 作者X
组件(e) Ver .1.3 作者Z
22
221a 用户方软件B
组件(b) Ver .1.1 作者X
组件(c) Ver .1.2 作者Y
222
23
231
用户方软件C
组件(d) Ver .1.1 作者Y
211
212
213
40
信息通知单元
30
注册更新单元
（关键部分和
补充部分的判定）
X
不更新
不向用户方软件通知
更新组件（b）的补充部分的请求
（从Ver.1.0到Ver.1.1）
（用户方软件B的更新者Y）

图6

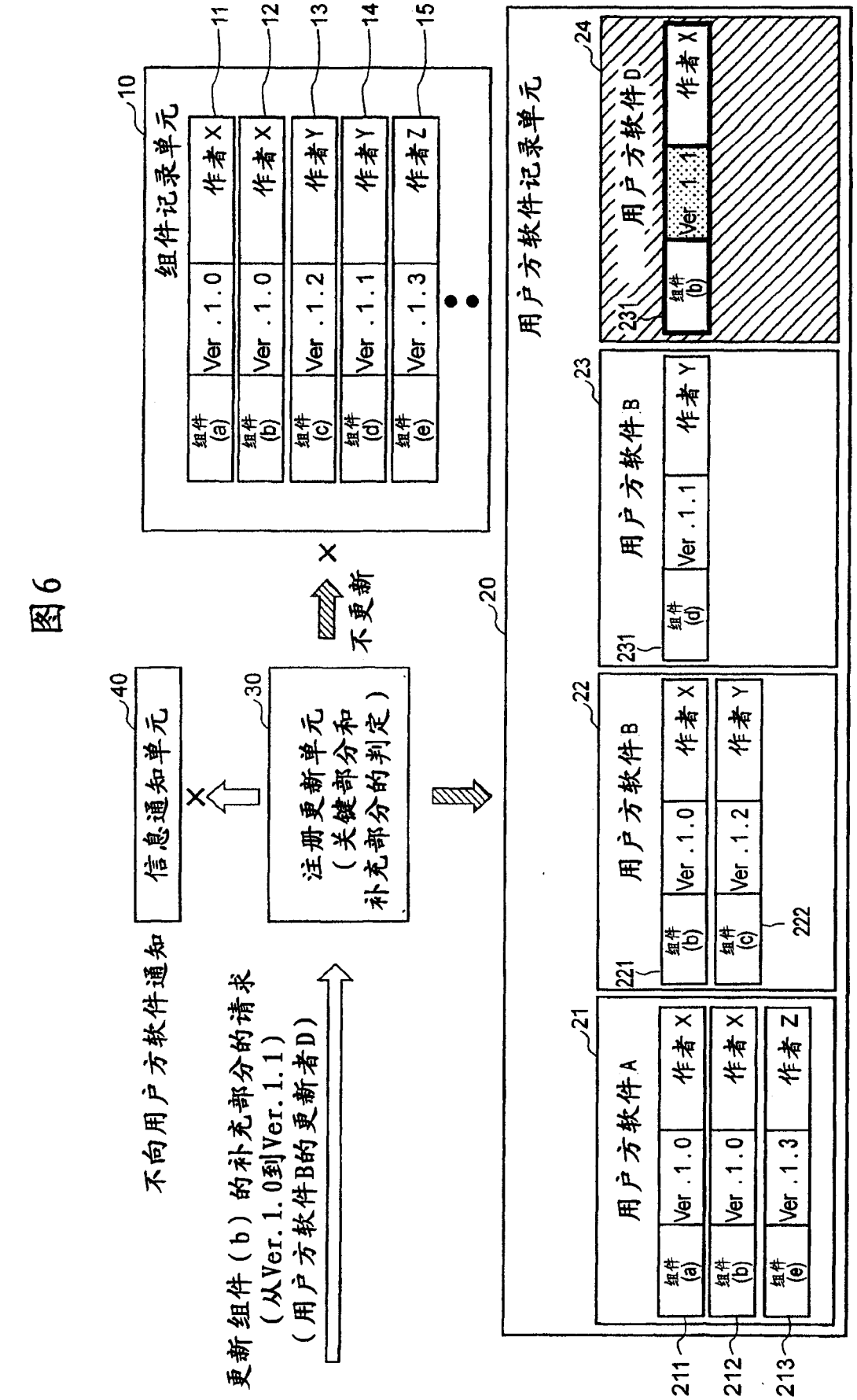
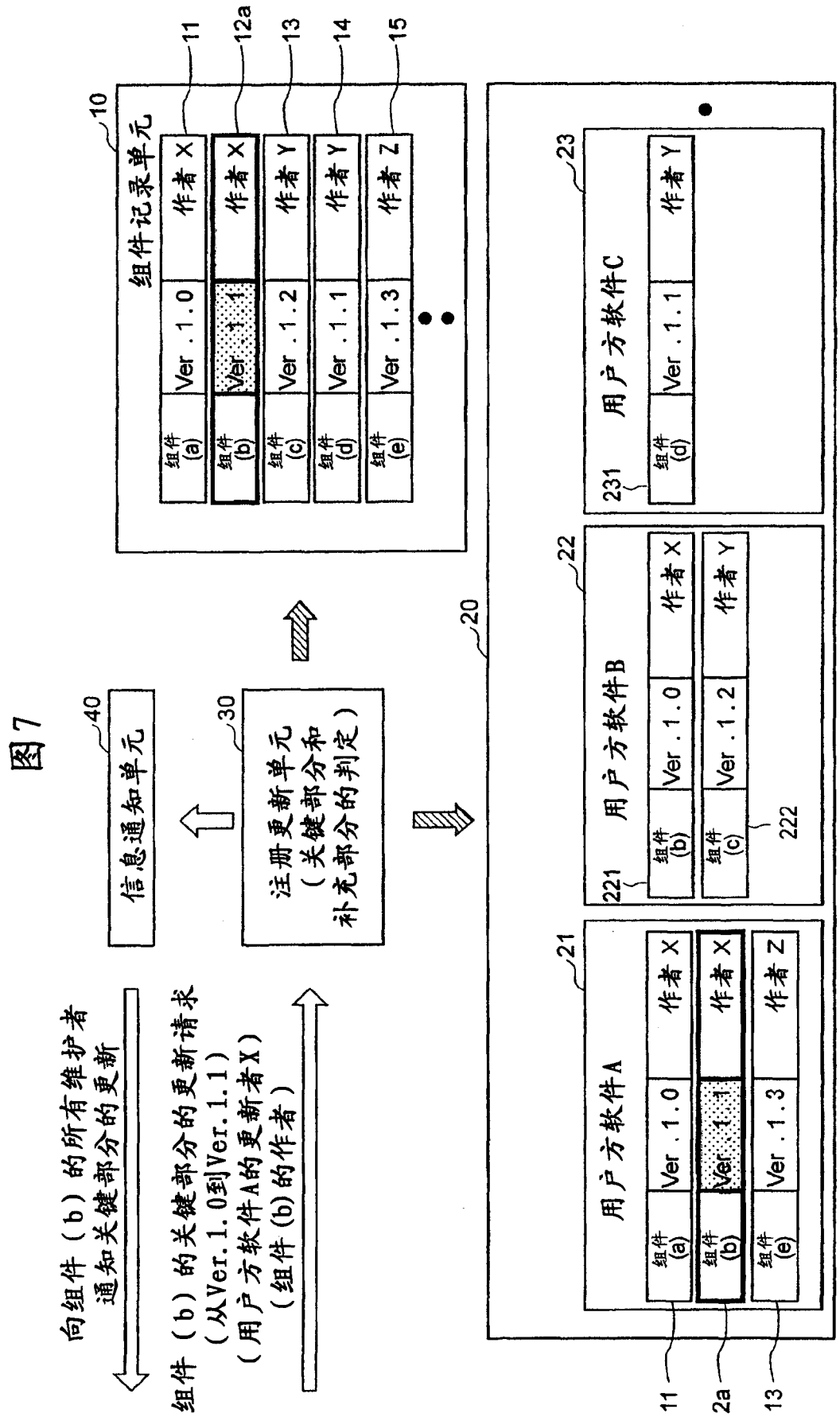

组件记录单元
组件(a) Ver .1.0 作者X
组件(b) Ver .1.0 作者X
组件(c) Ver .1.2 作者Y
组件(d) Ver .1.1 作者Y
组件(e) Ver .1.3 作者Z
10
11
12
13
14
15
不更新
注册更新单元
（关键部分和
补充部分的判定）
信息通知单元
不向用户方软件通知
更新组件（b）的补充部分的请求
（从Ver.1.0到Ver.1.1）
（用户方软件B的更新者D）
用户方软件记录单元
20
用户方软件A
组件(a) Ver .1.0 作者X
组件(b) Ver .1.0 作者X
组件(e) Ver .1.3 作者Z
21
211
212
213
用户方软件B
组件(b) Ver .1.0 作者X
组件(c) Ver .1.2 作者Y
221
222
22
用户方软件B
组件(d) Ver .1.1 作者Y
231
23
用户方软件D
组件(b) Ver .1.1 作者X
231
24

图7


向组件（b）的所有维护者
通知关键部分的更新
组件（b）的关键部分的更新请求
（从Ver.1.0到Ver.1.1）
（用户方软件A的更新者X）
（组件(b)的作者）
信息通知单元
40
注册更新单元
（关键部分和
补充部分的判定）
30
20
组件记录单元
10
组件(a) Ver .1.0 作者X 11
组件(b) Ver .1.1 作者X 12a
组件(c) Ver .1.2 作者Y 13
组件(d) Ver .1.1 作者Y 14
组件(e) Ver .1.3 作者Z 15
用户方软件A 21
组件(a) Ver .1.0 作者X
组件(b) Ver .1.1 作者X
组件(e) Ver .1.3 作者Z
用户方软件B 22
221
组件(b) Ver .1.0 作者X
组件(c) Ver .1.2 作者Y
222
用户方软件C 23
231
组件(d) Ver .1.1 作者Y
11
2a
13

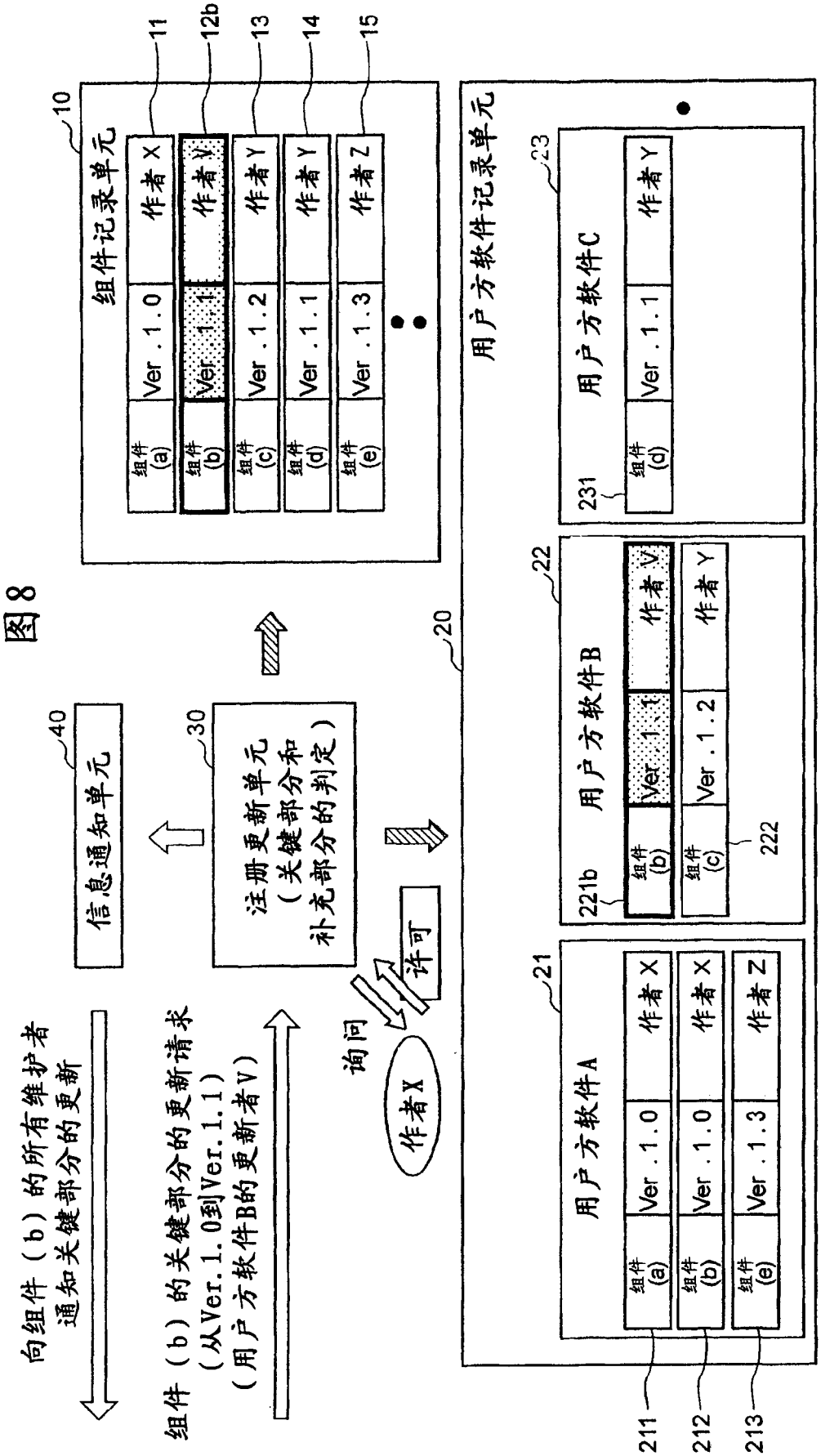
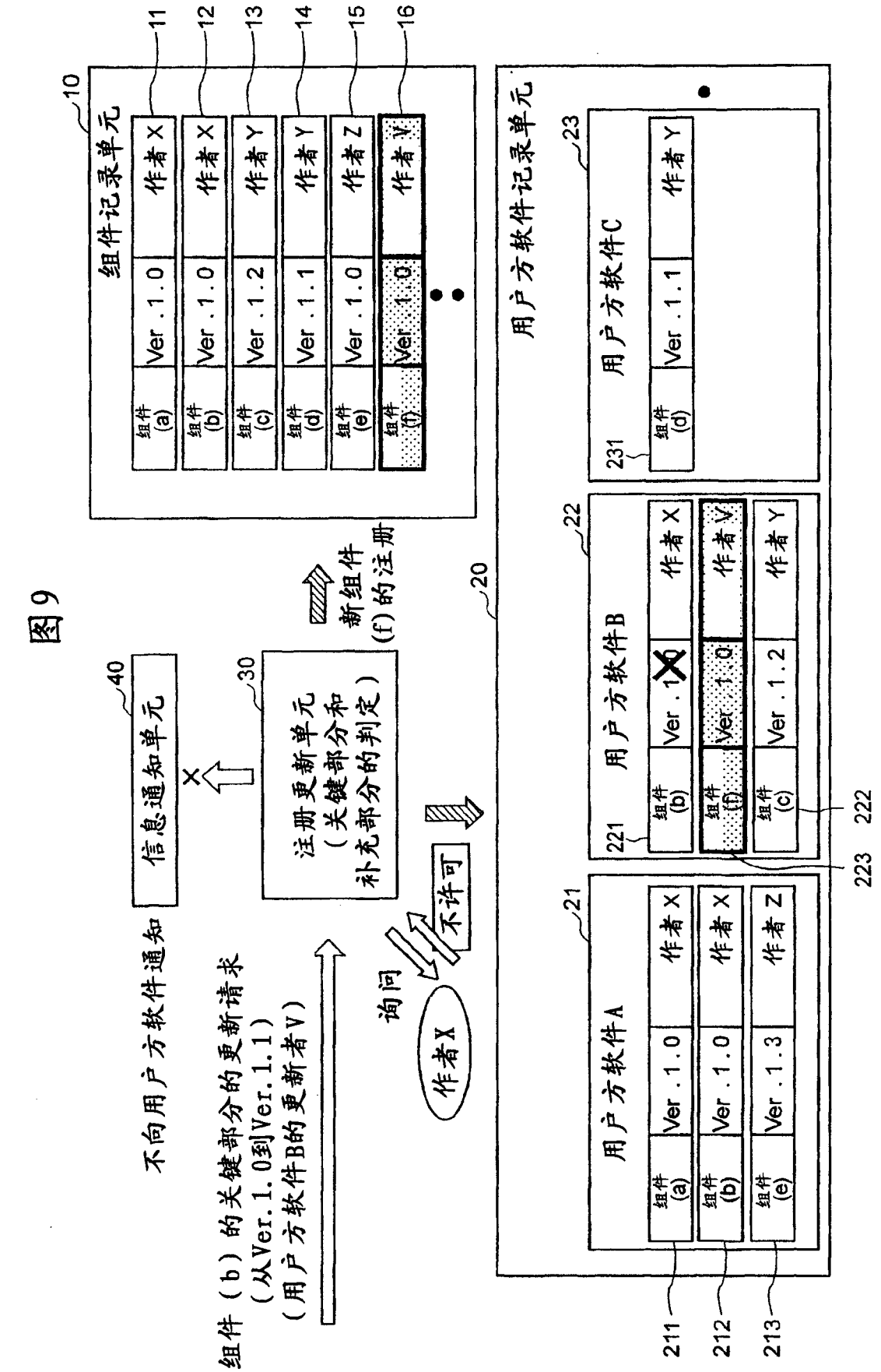


图 8
10
组件记录单元
11
12b
13
14
15
组件(a) Ver .1.0 作者X
组件(b) Ver .1.1 作者V
组件(c) Ver .1.2 作者Y
组件(d) Ver .1.1 作者Y
组件(e) Ver .1.3 作者Z
20
用户方软件记录单元
21
用户方软件A
组件(a) Ver .1.0 作者X
组件(b) Ver .1.0 作者X
组件(e) Ver .1.3 作者Z
211
212
213
22
用户方软件B
221b
组件(b) Ver .1.1 作者V
组件(c) Ver .1.2 作者Y
222
23
用户方软件C
231
组件(d) Ver .1.1 作者Y
40
信息通知单元
30
注册更新单元
（关键部分和
补充部分的判定）
向组件（b）的所有维护者
通知关键部分的更新
组件（b）的关键部分的更新请求
（从Ver.1.0到Ver.1.1）
（用户方软件B的更新者V）
询问
许可
作者X

图 9


10
组件记录单元
组件(a) Ver .1.0 作者X
组件(b) Ver .1.0 作者X
组件(c) Ver .1.2 作者Y
组件(d) Ver .1.1 作者Y
组件(e) Ver .1.0 作者Z
组件(f) Ver .1.0 作者V
11
12
13
14
15
16
20
用户方软件记录单元
21
用户方软件A
组件(a) Ver .1.0 作者X
组件(b) Ver .1.0 作者X
组件(e) Ver .1.3 作者Z
22
用户方软件B
221
组件(b) Ver .1 X 作者X
组件(f) Ver .1.0 作者V
组件(c) Ver .1.2 作者Y
222
223
23
用户方软件C
231
组件(d) Ver .1.1 作者Y
211
212
213
新组件(f)的注册
注册更新单元
(关键部分和补充部分的判定)
30
信息通知单元
40
不向用户方软件通知
组件(b)的关键部分的更新请求
(从Ver.1.0到Ver.1.1)
(用户方软件B的更新者V)
询问
作者X
不许可