[19] 中华人民共和国国家知识产权局

[51] Int. Cl[7]
G06F 9/44
B25J 9/00

[12] 发明专利说明书

[21] ZL 专利号 00800033.6

[45] 授权公告日 2005 年 5 月 4 日

[11] 授权公告号 CN 1200345C

[22] 申请日 2000.1.13 [21] 申请号 00800033.6
[30] 优先权
[32] 1999. 1.13 [33] JP [31] 6491/1999
[32] 1999. 1.13 [33] JP [31] 6757/1999
[32] 2000. 1.11 [33] JP [31] 6032/2000
[86] 国际申请 PCT/JP2000/000123 2000.1.13
[87] 国际公布 WO2000/042504 日 2000.7.20
[85] 进入国家阶段日期 2000.9.13
[71] 专利权人 索尼公司
地址 日本东京都
[72] 发明人 藤田雅博 坂本隆之 佐部浩太郎 高木刚
审查员 蔡 萍

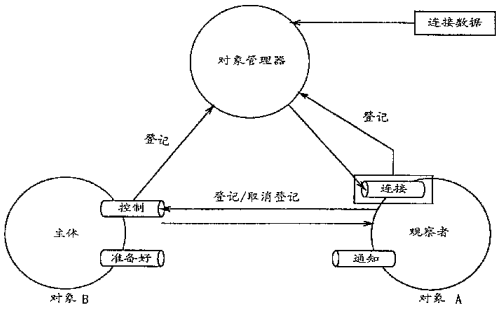
[74] 专利代理机构 北京市柳沈律师事务所
代理人 黄小临

权利要求书 4 页 说明书 33 页 附图 50 页

[54] 发明名称 算术处理装置,对象对对象的通信方法和机器人装置

[57] 摘要

本发明涉及一种算术处理装置、对象对对象的通信方法和一种机器人装置，本发明的目的在于，通过把本发明用于例如宠物型机器人装置上，可以简单地设置适合于当前配置的合适的软件。并且和过去相比，改善了对象的独立性。本发明根据相应于在每种配置中保存的部件检测结果的信息和与当前配置有关的部件检测结果的比较结果，按照所述配置改变软件。此外，本发明如此连接对象，使得可以根据连接数据进行数据交换。



ISSN1008-4274

1、一种用于在对象之间通信连接数据的算术处理装置，所述对象是包含在所述算术处理装置中的软件模块，所述算术处理装置包括：

对象管理器，用于连接所述对象，以便可以在所述对象之间交换所述连接数据；

部件检测装置，用于检测所连接的可配置物理部件并用于输出可配置物理部件检测结果；

信息存储装置，用于预先存储可配置物理部件连接信息；和

连接数据提供装置，用于比较包含于所述可配置物理部件检测结果中的可配置物理部件连接信息和所述预先存储的可配置物理部件连接信息，并基于所述比较结果向所述对象管理器提供连接所述对象所需的连接数据，

其中所述连接数据是用于指定对应的可配置物理部件和用于重新构建对象之间的通信所需要的数据。

2、如权利要求 1 所述的算术处理装置，其特征在于：

所述连接数据提供装置在从记录有多个连接数据的文件中选择连接数据之后向所述对象管理器提供连接数据。

3、如权利要求 1 所述的算术处理装置，其特征在于：

所述对象管理器基于来自所述连接数据提供装置的所述连接数据加载和卸载所述对象，并且同时重新构建所述对象之间的通信。

4、如权利要求 3 所述的算术处理装置，其特征在于：

所述对象管理器还改变所述算术处理装置的电功率消耗。

5、如权利要求 1 所述的算术处理装置，其特征在于：

所述连接数据利用文本数据描述。

6、一种用于在对象之间通信数据的对象对对象的通信方法，所述对象包含在一算术处理装置中，其特征在于所述方法包括下列步骤：

检测连接到所述算术处理装置的可配置物理部件并输出可配置物理部件检测结果；

比较包含于所述可配置物理部件检测结果中的可配置物理部件连接信息和预先存储的可配置物理部件连接信息；

基于所述比较结果向对象管理器提供连接所述对象所需的连接数据；以及

连接所述对象，使得可以在所述对象之间交换所述连接数据，

其中所述连接数据是用于指定对应的可配置物理部件和用于重新构建对象之间的通信所需要的数据。

7、如权利要求 6 所述的对象对对象的通信方法，其特征在于：

所述连接数据从记录有多个连接数据的文件中选择并被提供。

8、如权利要求 6 所述的对象对对象的通信方法，其特征在于：

基于来自所述连接数据提供装置的所述连接数据加载和卸载所述对象，并且同时重新构建所述对象之间的通信。

9、如权利要求 8 所述的对象对对象的通信方法，其特征在于：

改变所述算术处理装置的电功率消耗。

10、如权利要求 6 所述的对象对对象的通信方法，其特征在于：所述连接数据是用文本数据描述的。

11、一种机器人装置，其特征在于：所述机器人装置的软件模块在其可配置物理部件被交换之后可以被改变；以及整个操作被由对象构成的软件通过在各对象之间交换连接数据进行控制；并且包括：

对象管理器，用于连接所述对象，使得可以在所述对象之间交换所述连接数据；

部件检测装置，用于检测所连接的可配置物理部件并用于输出可配置物理部件检测结果；

信息存储装置，用于预先存储可配置物理部件连接信息；和

连接数据提供装置，用于比较包含于所述可配置物理部件检测结果中的可配置物理部件连接信息和所述预先存储的可配置物理部件连接信息，并基于所述比较结果向所述对象管理器提供连接所述对象所需的连接数据，

其中所述连接数据是用于指定对应的可配置物理部件和用于重新构建对象之间的通信所需要的数据。

12、如权利要求 11 所述的机器人装置，其特征在于：

所述连接数据从记录有多个连接数据的文件中选择并被提供。

13、如权利要求 11 所述的机器人装置，其特征在于：

所述对象管理器基于来自所述连接数据提供装置的所述连接数据加载和卸载所述对象，并且同时重新构建所述对象之间的通信。

14、如权利要求 13 所述的机器人装置，其特征在于：

所述对象管理器还改变所述算术处理装置的电功率消耗。

15、如权利要求 11 所述的机器人装置，其特征在于：所述连接数据是用文本数据描述的。

16、一种具有由多个对象组成的软件系统的机器人设备，包括：

连接信息输出装置，用于根据连接于所述机器人设备的机身的所述机器人设备的各部件输出连接信息；

文件存储装置，用于存储文件，对于根据所述机器人设备的所述部件的连接而改变的所述机器人设备的每种形式，所述文件都描述了所述机器人设备的至少一个类型的配置；

连接文件存储装置，用于对所述机器人设备的每种形式存储连接文件，所述连接文件记录有相应的对象和在各个对象之间通信所需的连接数据；

形式指定装置，用于通过比较所输出的连接信息与描述机器人设备的配置的所述文件，指定所述机器人设备的当前形式；以及

用于根据所指定的形式访问所述连接文件，并通过根据所述形式重新构建所述软件系统的对象、基于所述机器人设备的形式来更新软件的装置。

17、根据权利要求 16 所述的机器人设备，其中对于所述机器人设备的每种形式，所述连接文件存储装置存储记录有相应的对象和在各个对象之间通信所需的数据的连接数据以及标记，并且所述机器人设备还包括：

标记检测装置，用于根据由所述形式指定装置指定的所述形式检测与当前形式对应的所述标记；以及

用于从所检测的标记访问所述连接文件，并通过根据对应于所述标记的所述连接数据重新构建所述软件系统的对象、基于所述机器人设备的形式来更新软件的装置。

18、根据权利要求 17 所述的机器人设备，其中在所述标记中至少描述要使用的数据格式和对象名称。

19、一种应用于具有由多个对象组成的软件系统的机器人设备的软件更新方法，包括：

连接信息输出步骤，用于根据连接于所述机器人设备的机身的所述机

器人设备的各部件输出连接信息;

形式指定步骤，用于比较所输出的连接信息与一文件，来指定所述机器人设备的当前形式，对于根据所述机器人设备的所述部件的连接而改变的所述机器人设备的每种形式，所述文件都描述了所述机器人设备的至少一个类型的配置；以及

用于根据所指定的形式访问连接文件，并通过根据所述形式重新构建所述软件系统的对象、基于机器人设备的形式来更新软件的步骤，对于所述机器人设备的每种形式，所述连接文件记录有相应的对象和在各个对象之间进行通信所需的连接数据。

20、根据权利要求 19 所述的软件更新方法，其中对于所述机器人设备的每种形式，还记录有相应的对象和在各个对象之间通信所需的数据的连接数据以及标记，并且所述软件更新方法还包括:

标记检测步骤，用于根据由所述形式指定步骤指定的所述形式检测与当前形式对应的所述标记；以及

用于从所检测的标记访问所述连接文件，并通过根据对应于所述标记的所述连接数据重新构建所述软件系统的对象、基于所述机器人设备的形式来更新软件的步骤。

21、根据权利要求 20 所述的软件更新方法，其中在所述标记中至少描述要使用的数据格式和对象名称。

算术处理装置，对象对对象的通信方法和机器人装置

技术领域

本发明涉及算术处理装置、对象对对象的通信方法和机器人装置，并且可以用于例如宠物型机器人装置。本发明可以根据在每个配置中存储的相当于部件检测结果的信息和关于当前配置的部件检测结果的比较的结果，通过相应于配置改变一软件来简单地设置相应于当前配置的合适的软件。此外，通过这样连接各对象，使得可以根据连接数据进行数据交换，和过去相比，本发明可以增加对象的独立性。

背景技术

以前，例如在宠物型机器人装置中，形成面向对象的软件，并且各种数据按照对象之间的通信进行处理，因而机器人装置可以进行例如类似于动物的运动。

不过，在这种宠物型机器人装置中，据认为如果其中的部件，例如腿，通过按照需要交换这些部件，能够按照用户所需的配置设置，则这种宠物型机器人装置可以成为更通用的。

此外，在这种情况下，希望能够按照配置简单地交换软件。

然而，在常规的面向对象的软件中，因为软件是在对象之间的明确的关系的预定条件下设计的，所以对象彼此之间高度地相关。因而，即使软件能够被改变，从而和相对自由的配置的改变相适应，结果，仍需要开发和每种配置兼容的特殊的软件。

作为用于摆脱对象之间的这种依赖性的方法，可以考虑通过使用例如使用设计模式(design pattern)(观察者(observer)模式)的方法进行软件开发的方法(Erich Gamma er al.，用于重用的面向对象设计图形，第313-324页)。更具体地说，在按照这种设计模式的方法中，因为其中对对象分类为主体和观察者，并在主体上登记(register)观察者，所以对象根据所述登记彼此相连，并且各种信息和数据从观察者传递给主体。

在这种情况下，主体从被记录的观察者中连续调用指定(更新)的方法，并把数据处理提交给观察者。因而，在这种设计模式方法中，当进行设计时，可以高度独立地对主体进行设计，而不需要编制诸如输出数据给特定的对象的代码。

然而，即使在这种方法中，也需要观察者被记录在特定的主体上，并且观察者应当预选被描述。因而，实际上，按照这种设计模式的方法在保证每个对象的独立性方面是不够的。

在另一方面，提出了一种通过利用这种方法形成视觉工具(visual tool)来增加对象的独立性的方法(JIP对象技术研究协会，视觉时代C++导论，第182-185页)。然而，按照这种方法，需要进行编译，以便通过在用视觉工具指定对象的连接之后，自动地形成源代码来改变软件结构。

此外，可以考虑通过使用代理在对象之间进行通信的方法，如日本专利申请公开10-28398中提出的方法。然而，这种方法不能增加对象的独立性。

此外，可以考虑利用客户机服务器系统的命名服务的方法，例如日本专利公报10-171465中披露的。更具体地说，在这种客户机服务器系统中，应用与观察者模式(observer pattern)比较的软件打包方法。并且这种方法也在CORBA(Common Object Request Broker Architecture公用对象请求中介架构)中采用了。

然而，按照这种方法，实际上，仍然不足以增加对象的独立性，因为观察者如同在观察者模式的情况下一样指定主体，并且仍然需要再编译和再链接。

发明内容

本发明便是考虑上述要求作出的，首先，本发明提出一种能够按照当前的配置简单地设置合适的软件的机器人装置。

此外，在作为特殊要求在交换部件例如腿并按照用户所需的形式进行设置的情况下，据认为对象应当被重新构成，并且软件应当相应于软件中的配置的改变而改变，以便控制整个操作。

作为本发明的第二方面，提出一种算术处理装置、对象对对象的通信方法和机器人装置，和传统的装置相比，它们能够改善对象的独立性。

为了克服这种问题，为所述机器人装置提供：部件检测装置，用于检测和机器人装置相连的部件，并输出部件检测结果；信息存储装置，用于存储在通过交换部件而要获得的每种配置中的相应于部件检测结果的信息；以及软件改变装置，用于根据部件检测结果和在信息存储装置中存储的信息的比较结果，按照所述配置改变软件。

为了解决这种问题，为所述算术处理装置提供：对象管理器，用于以这样的方式连接对象，使得可以交换数据；以及连接数据提供装置，用于向对象管理器提供用于连接对象所需的连接数据。

此外，提供一种在对象之间的通信方法，其中根据从预定的数据提供装置要被提供的连接数据以这样的方式连接对象，使得数据可以被交换。

此外，为机器人装置提供：对象管理器，用于以这样的方式连接对象，使得可以交换数据；以及连接数据提供装置，用于向对象管理器提供用于连接各对象所需的连接数据。

此外，根据通过检测被连接的部件而要获得的部件检测结果和相应于在交换部件时获得的每种配置中的部件检测结果的数据的比较结果，通过相应于所述配置改变软件，软件可以相应于在改变部件之后的配置的改变而改变。因而，可以简单地设置相应于当前配置的合适的软件。

因为提供有对象管理器，用于连接各对象以便可以进行数据交换；以及连接数据提供装置，用于向对象管理器提供用来连接对象所需的连接数据，所以在设计连接对象时，可以不考虑任何特殊的连接对象，并且，对于这样设计的对象，连接数据可以用各种方式设置和改变。因此，和传统的装置相比，对象的独立性被显著地改善。

此外，通过以这样的方式连接各对象，使得可以根据由预定的数据提供装置提供的连接数据进行数据交换，在设计对象时可不考虑任何被连接的特定对象。对于这样设计的对象，连接数据可以用各种方式设置，并且可以用各种方式改变连接，因而，和过去的装置相比，对象的独立性被显著地改善。

此外，通过提供对象管理器，用于以这样的方式连接对象，使得可以进行数据交换；以及连接数据提供装置，用于向对象管理器提供用来连接各对象所需的连接数据，使得在设计每个对象时可不考虑任何特定的要被连接的对象。此外，对于这样设计的对象，连接数据可以用各种方式设置，

并且可以用各种方式改变连接，因而，和传统的装置相比，对象的独立性被显著地改善。

本发明还提供一种具有由对象组成的软件系统的机器人设备，包括：连接信息输出装置，用于根据连接于所述机器人设备的机身的所述机器人设备的各部件输出连接信息；连接模板存储装置，用于存储连接模板，所述连接模板中对于根据所述机器人设备的所述部件的连接改变的每种形式的所述机器人设备都描述了所述机器人设备的至少一个类型的配置；形式指定装置，用于指定比较所述连接信息与所述连接模板的所述机器人设备的当前形式；以及用于通过根据由所述形式指定装置指定的所述机器人设备的形式重新构建所述软件系统的对象，更新基于所述机器人设备的所述形式的软件。

本发明还提供一种控制对象之间通信的处理器，包括：多个定义输入和输出的对象；信息存储装置，用于存储所述每个对象之间的输入和输出的连接关系信息；和控制装置，用于根据所述连接关系信息在规定的时间连接所述对象。

本发明还提供一种应用于具有由对象组成的软件系统的机器人设备的软件更新方法，包括：连接信息输出步骤，用于根据连接于所述机器人设备的机身的所述机器人设备的各部件输出连接信息；形式指定步骤，用于指定比较所述连接模板与所述连接信息的所述机器人设备的当前形式，所述连接模板中对于根据所述机器人设备的所述部件的连接改变的每种形式的所述机器人设备都描述了所述机器人设备的至少一个类型的配置；以及用于通过根据由所述形式指定装置指定的所述机器人设备的形式重新构建所述软件系统的对象，更新基于所述机器人设备的所述形式的软件。

本发明还提供一种控制对象之间通信的控制方法，其中：根据所述每个对象之间的输入和输出的连接关系信息，在规定的时间连接定义了输入和输出的所述各对象。

附图说明

图 1 是表示按照本发明的机器人装置的结构的透视图；

图 2 是表示图 1 的机器人装置的结构的方块图；

图 3 是表示四腿型机器人装置中的软件的结构示意图；

图 4 是表示轮型机器人装置中的软件结构的示意图;

图 5 是表示四腿型机器人装置中姿势改变的状态转换图;

图 6 是表示输入和状态之间的关系的图;

图 7 是表示设计机器人装置的处理过程的流程图;

图 8 表示设计文件的描述;

图 9 表示连接数据;

图 10 表示对象管理器、虚拟机器人装置和设计的机器人装置启动之后的操作的时间图;

图 11 是表示对象管理器、虚拟机器人装置和设计的机器人装置在插入和拔出时的操作的时间图;

图 12 是表示观察者和主体之间的关系的示意图;

图 13 是表示在多个观察者中观察者和主体之间的关系的示意图;

图 14 表示在多个主体中观察者与主体之间的关系的示意图;

图 15 是表示对象管理器和观察者以及主体之间的关系的示意图;

图 16 是表示在接收连接数据的情况下对象管理器的操作的时间图;

图 17 是在 DoStart 之后程序的时间图;

图 18 是表示在插入和拔出以及状态改变时的程序的时间图;

图 19 是说明连接数据和观察者以及主体之间的关系的示意图;

图 20 表示相关存储器 1 的描述;

图 21 表示相关存储器 2 的描述;

图 22 是表示按照本发明的第二实施例的轮胎型机器人装置的软件结构的示意图;

图 23 是表示对象的连接的示意图;

图 24 是表示关于结构变化的连接条件的改变的示意图;

图 25 是表示在图 22 中的机器人装置的启动时处理步骤的流程图;

图 26 是表示图 25 的处理中初始化的解释的示意图;

图 27 是表示在图 25 的处理中启动的解释的示意图;

图 28 是从图 25 继续的处理步骤的定时图;

图 29 是说明图 28 的处理中对象的加载的示意图;

图 30 是表示从图 28 继续处理的定时图;

图 31 是在图 30 的处理中的连接的示意图;

图 32 表示服务管理器中的数据;

图 33 是表示从图 31 继续处理的示意图;

图 34 是表示服务管理器的处理的示意图;

图 35 是说明各对象的连接的示意图;

图 36 是说明和图 19 相比在观察者和主体之间的关系和连接数据的示意图;

图 37 是表示和图 28 相比每个对象的连接的示意图;

图 38 是说明在图 30 中的启动的示意图;

图 39 是表示图 22 的机器人装置中的在改变配置(configuration)时的处理的定时图;

图 40 是表示在图 39 的处理中的配置改变的通知的示意图;

图 41 是表示在图 39 的处理中的停止处理的示意图;

图 42 是表示继续图 39 的处理的定时图;

图 43 是表示在图 42 的处理中的断开对象的示意图;

图 44 是表示继续图 43 的处理的示意图;

图 45 是表示继续图 42 的处理的定时图;

图 46 是表示和配置改变有关的对象的改变的示意图;

图 47 是表示在配置改变之后的每个对象的关系的示意图;

图 48 是表示服务管理器的处理的示意图;

图 49 是说明通过复制在各对象之间通信的示意图;

图 50 是说明通过公用存储器在各对象之间通信的示意图;

图 51 是说明公用存储器监视的示意图;

图 52 是说明其中有多个数据被编辑的数据通信的示意图;

具体实施方式

下面参照附图说明本发明的实施例。

(1)第一实施例的结构

(1-1)总体结构

图 1 是表示按照本发明的实施例的机器人装置透视图。机器人装置 1 是一种所谓的宠物型机器人装置，其具有狗的形状，并且能够利用四条腿行走。这种机器人装置 1 的主单元 2 和可动单元 3 相连，并且主单元 2 包

括头单元 2A 和身体单元 2B。

头单元 2A 配备有麦克风 4、触觉传感器 5、具有立体视觉的电视镜头 6、遥控红外线接收单元 7，并且由这些单元可以获得各种数据。此外，头单元 2A 配备有图像显示单元 9 和扬声器 10，并且通过这些装置可以发出各种数据。

操作器 11 被设置在身体单元 2B 的背部上，天线被设置在相应于尾巴的位置，通过天线可以接收用户的操作。此外，身体单元 2B 具有一个槽，在其中可以插入 IC 卡，例如可以进行软件的版本升级。

此外，身体单元 2B 配备有控制器 15，用于通过处理借助于头单元 2A 输入和输出的各种信息和例如来自操作器 11 的操作信息控制整个操作；电源单元 14，用于向各个单元提供电源；通信单元 16，用于通过天线 12 接收和发送各种信息。并且在其下部设置有电池 17。

在另一方面，可动单元 3 配备有四条腿，在相应于关节的位置设置有致动器和角度检测传感器，并通过串行总线和主单元 2 相连，并可以通过主单元 2 的控制进行操作。这样，机器人装置 1 便可以利用四只腿运动。

此外，该可动单元 3 被如此设置，使得其后腿可以和主体 2 脱离。具有轮胎的可动单元可以和机器人装置 1 相连，用于代替后腿。利用这种结构，机器人装置 1 的配置可以按照用户的选择在利用四条腿运动的配置(以后称为四腿型)和利用轮胎运动的结构(以后称为轮胎型)之间改变。

图 2 是表示所述机器人装置 1 的结构的方块图。在机器人装置 1 中，记录有处理程序的程序介质 20 通过外围设备 21 和中央处理单元 22 相连，并且在程序媒体 20 中记录的处理程序将由所述中央处理单元 22 执行。此外，中央处理单元 22 通过外围设备 21 与每个致动器、运动单元 3 的传感器、电视摄像头 6 以及机器人装置部件 24，25，26（即，操作器 11）相连，因而利用这个结构 CPU 22 可以控制整个操作。

此外，中央处理单元 22 通过外围设备 21 和构成电源单元 14 的用于电源控制的单片微计算机 27 相连。因而控制所述单片微计算机 27 的操作的 CPU 22 从电池 17 向所有单元提供电源。还有，CPU 22 在节能方式下不从电池 17 而是从钮扣电池 29 向所有的单元提供电源。

(1-2)软件构成

图 3 表示借助于利用分层的四腿型机器人装置 1 中的软件结构的示意

图。图 4 是和图 3 相对照的轮型机器人装置中的软件结构的示意图。在这种软件结构中，最低层是设备驱动器层，其中设置有各个设备的驱动器。在这里，设备(device)驱动器在传感器、运动单元 3 中的致动器以及被设置在头部单元 2A 的各个机器人装置部件之间输入和输出各种数据，并提供用于这些处理的软件对象。

电池管理设备驱动器(电池管理器 DD)通过和电源单元 14 的单片机 27 进行数据通信检测电池 17 的剩余电量，并周期性地把检测结果通知上级的机器人装置系统软件。此外，电池管理器控制中央处理单元 22 的时钟，并按照机器人装置系统软件的规格减少所述时钟的频率，停止设备的不需要的操作，因而减少电源的总的消耗量。

此外，机器人装置串行总线驱动器(机器人装置串行总线 DD)接收和串行总线相连的运动单元 3 的传感器信息，并通过与设置在主单元 2 上的串行总线的主控制硬件进行数据通信，把所述信息通知给上级的机器人装置系统软件。与此相反，其向每个设备输出要从机器人装置系统软件输出的控制数据和音频信号。

此外，机器人装置串行总线设备驱动器当启动系统时检测和串行总线相连的设备的改变(添加或删除)。此外，接收和所述改变有关的设备专用信息，它向机器人装置系统软件发送表示机器人装置 1 的当前配置的信息。

机器人装置系统软件被设置在该设备驱动器层的上级。在这里，机器人装置系统软件由软件对象例如虚拟机器人装置、设计机器人装置、电源管理器和对象管理器构成。

虚拟机器人装置在和机器人装置串行总线设备驱动器进行数据通信时，把具有每个装置专用格式的数据转换成机器人装置公用格式，并且进行接收和发送这些数据。更具体地说，关于具有设备专用格式的电机控制的数据，电位器传感器输出用借助于模数转换获得的 10 位表示。在另一方面，在虚拟机器人装置中，被这样表示的数据被转换成最低级的 1 位是 0.001°的数据，并以机器人装置 1 中的公用格式输出。此外，虚拟机器人装置，除了通过这种格式转换进行数据接收和发送之外，还向上级软件发送由电视摄像头 6 获得的图像数据。

此外，虚拟机器人装置从机器人装置串行总线设备驱动器接收表示机器人装置 1 的当前配置的信息，并编译所述信息。因而，虚拟机器人装置

控制表示机器人装置部件的种类和这些部件连接方式和连接顺序的(CPC(可配置的物理部件)连接部件)的连接信息，并把所述 CPC 连接信息通知设计机器人装置。

此外，当在和串行总线相连的设备中发生改变时，虚拟机器人装置接收来自设备驱动器的机器人装置串行总线设备驱动器的通知，并把其通知给设计机器人装置。

当收到来自虚拟机器人装置的 CPC 连接信息时，设计机器人装置，便连续地比较这些信息和预先存储的连接模板信息，选择适用于机器人装置 1 的当前配置的模板。此外，设计机器人装置指令对象管理器把上级的配置依赖软件更新为适合于当前配置的配置依赖软件。

在这种连接中，按照本实施例，连接模板信息将在设计文件中作为设计数据被描述。

对象管理器通过使用关于要被设计机器人装置通知的对象的信息把配置依赖软件更新为适合于当前配置的配置依赖软件，并按照设计机器人装置的命令更新每个对象的连接。更具体地说，对象管理器指令停止操作、断开对象之间的通信、破坏对象、释放构成配置依赖软件的所有对象上的资源。此外，对象管理器命令加载新需要的对象、进行初始化、并构建这些对象的连接和启动。在这种连接中，对象之间的连接意味着用于在各对象之间进行接收/发送数据的连接。

电源管理器在电池管理器设备驱动器之间通信交换数据，并指令电池管理器设备驱动器转换时钟，并且还按照上级软件(配置独立软件或配置依赖软件)的指令来指令对象的停止操作。

配置依赖软件被置于所述机器人装置系统软件的上级，配置独立软件被置于更上面的级。其中，配置依赖软件是按照机器人装置 1 的配置而改变的软件。与此相反，配置独立软件是固定的，和机器人装置的配置无关，并被置于机器人装置 1 的上级。因此，配置依赖软件根据所述机器人装置 1 的配置具有不同种类的对象。按照本实施例，一般说来，适用于四腿型的和轮型的配置的软件可以通过改变配置依赖软件而被简单地构成。

更具体地说，在四腿型中，所述配置依赖软件(图 3)由对象例如指令转换器、运动网络、运动重演(replay)、行走模式(walking pattern)发生器和与运动单元 3 相关的对象构成。

命令转换器用于把来自不依赖于配置的配置独立软件的命令转换为适用于机器人装置 1 的配置的命令。并且当从配置独立软件输入指令姿势(例如睡眠、休息、行走、运动)的命令作为与运动单元 3 有关的命令时，命令转换器把这些命令转换为用于规定姿势(例如睡眠、坐、站、行走)的命令。

按照图 5 所示的转变过程，运动网络按照表示要从命令转换器输出的这些姿势的命令来启动相应于在这个转变图中的箭头的对象方法(入口(entry))。更具体地说，当在睡眠状态下输入行走的命令时，运动网络便相继地启动运动重演(replay)对象的相应的入口，以便连续地把姿势改变为坐、站和行走。在这种情况下，当其达到最后的行走状态时，运动网络便启动相应于表示这一状态的其自身循环的行走模式发生器的入口。

当由运动网络进行的入口开始时，运动重演部分输出相应于这个姿势改变的关节角度命令值。为了更加具体，运动重演部分存储作为键控数据记录的离散的关节角度命令值，并通过利用所述键控数据进行内插算术处理来发送一系列角度命令值。

行走模式发生器计算每个关节角度命令值并输出所述的命令值，使得其向着由配置独立软件规定的方向运动。

视觉部分是一种例如用于图像处理的对象，其接收来自虚拟机器人装置对象的由电视摄像头 6 获得的图像数据并处理这些数据。因而，所述视觉部分识别例如红色的物体，并根据这个机器人装置 1 检测这个红色物体的位置和外形尺寸，并发送检测结果。

在另一方面，在轮型中，配置依赖软件(图 4)具有和四腿型相同的视觉对象和轮型专用对象，即命令转换器、运动命令器、运动重演部分(replay)2 和作为和运动单元 3 相关的对象的轮型模式发生器。

命令转换器和在四腿型中的命令转换器的情况下一样，用于把来自不依赖于配置的配置独立软件的命令转换为适用于机器人装置 1 的配置的命令。在这种情况下，因为机器人装置 1 是轮型的，所以命令转换器把用于规定睡眠、休息、醒来和运动的指令分别转换为睡眠、休息、准备和行进的命令。

运动命令器接收命令转换器的输出信号，构成和所述轮胎的控制命令相关的关节角度命令值。在输入睡眠命令的情况下，运动命令器设置关节

角度命令值为“中性”。这里中性状态意味着不被驱动或制动。因而，构成运动单元的两臂被保持在悬空状态。并且轮胎保持在停止状态。

在另一方面，在输入休息指令的情况下，运动命令器产生关节角度命令值，使得两臂在肘部向前弯曲 90 度，并且头部 2A 朝向其脸前方。但是轮胎保持在停止状态。

此外，当输入准备命令时，运动命令器产生关节角度命令值，使得两臂向前伸直肘关节，头单元 2A 朝向其脸前方。在这种情况下，轮胎也保持在停止状态。此外，当输入行进指令时，便发出一个指令值使得轮胎向前滚动，同时两臂保持在准备状态的情况下的状态。

运动重演部分 2 按照运动命令器的指令控制和轮子型特定的操作相关的除轮胎之外的部件的操作(例如两臂的操作)。轮子模式发生器按照运动指令器的控制产生轮胎的轮子控制信息。

配置独立软件是一种不依赖于机器人装置 1 的配置的软件，并具有配置设置对象。这里配置设置对象从配置独立软件接收不依赖于机器人装置 1 的配置的信息。并按照所述信息，其向配置依赖软件输出如上所述的表示机器人装置 1 的姿势(例如睡眠)的指令。

例如，如果根据红色物体的存在或不存在以及要由视觉对象提供的位置信息在 1 分钟的时间间隔内没有检测到红色物体的存在，则发出指令使当前状态处于睡眠状态。并且，如果处于休息状态，则该状态将被转换到睡眠状态，如果处于醒来状态，则被转换到休息状态，如果处于运动状态则被转换到醒来状态，如图 6 所示。

此外，在以小于预定值的一个较小的值(比机器人装置设置的值小 10)检测到红色物体的情况下，并且如果这种情况持续时间大于 10 秒钟，则配置设置对象发出指令，使得从睡眠状态转换为休息状态，从休息状态转换为醒来状态，从醒来状态转换为运动状态。此外，其发出指令使得在运动状态下继续运动状态。

此外，当在比预定值较大的值下检测到红色物体时，配置设置对象则发出指令，使得睡眠状态被转换为醒来状态，休息和醒来状态被转换为运动状态。此外，还发出指令，使得在运动状态下仍然保持运动状态。

因而，在机器人装置 1 中，最高级软件被这样构成，使得其不依赖于配置，并且按照配置独立格式在这些配置独立软件之间接收与/或发送数

据。利用这种结构，在机器人装置 1 中，只有配置依赖软件被按照配置改变，因而可以相应于配置简单地改变软件。

(1-3)配置依赖软件的改变

在机器人装置 1 中，这种配置依赖软件按照对象管理器的处理用设计机器人装置的指令进行更新。此时，按照本实施例，配置依赖软件将通过具有配置依赖软件的对象的加载和卸载并通过重构用于在对象之间进行通信的连接而被更新。

因而，设计机器人装置接收来自虚拟机器人装置的规定机器人装置 1 的配置的 CPC 连接信息，并且所述设计机器人装置根据所述 CPC 连接信息把配置依赖软件的更新引导到对象管理器，并提供所需的信息。

此外，虚拟机器人装置含有通过在低层上的机器人装置串行总线设备驱动器和串行总线相连的每个装置的信息，并由这一信息产生 CPC 连接信息，并将所述信息通知到设计机器人装置。

因而，在所述机器人装置 1 中，预定的信息被记录在每个机器人装置部件的存储器上，并按照要通过机器人装置串行总线设备驱动器执行的数据通信，可以获得具有其在虚拟机器人装置 1 中的位置信息的和串行总线相连的设备的信息。

机器人装置 1 包括电视摄像头，扬声器，麦克风以及各种致动器和与串行总线相连的传感器，这些是构成机器人装置的 CPC 基本部件。并且每个模型(CPC 模型)由在连接的状态下设置的这些 CPC 基本部件的多种组合构成。例如，一只腿，它是一个型号(model)，由具有 3 个电机和一个开关的多个基本部件的特定的连接状态组成。

在机器人装置 1 中，对每个模型提供特定的识别数据，因而每个和串行总线相连的部件可以通过这些识别数据被识别。此外，这种 ID 数据包括制造模型的制造厂的 ID，和相应于该型号的 ID。此外，在模型中的每个部件信息和每个部件的位置信息按照每个部件的连接信息和在模型中的每个部件的类属信息被设置。并且这些信息连同每个模型的 ID 数据被存储在每个模型的存储器中。在这种连接中，按照本实施例，如果在模型的存储器容量小，则在每个模型中只记录 ID 数据，在模型中的各部件的部件信息和位置信息被存储在主单元 2 中。

当系统启动或者设备被改变时，形成 CPC 连接信息的虚拟机器人装置

把连接信息通知给设计机器人装置，所述连接信息用 CPC 基本部件和表示每个部件的连接顺序的数据结构来形成，所述部件连接顺序根据 ID 数据和从机器人装置串行设备驱动器提供的位置信息连续地跟踪具有树结构的串行总线来确定。因而，要从虚拟机器人装置提供的 CPC 连接信息由相应于所述配置的 CPC 基本位置信息构成。

设计机器人装置参照设计文件，根据要从虚拟机器人装置提供的 CPC 连接信息，选择相应于当前机器人装置 1 的配置的连接模板(其标记在后面说明)。此外，设计机器人装置按照所选择的模板指令对象管理器把其更新为适用于当前配置的配置依赖软件。

在这里，设计文件是文本数据格式的描述，并通过在机器人装置 1 的每种配置中的每个模型的 CPC 基本(primitive)位置信息组中附加标记来进行描述。此外，CPC 基本位置信息包括每个模型的 CPC 基本信息，和每个部件的 CPC 坐标定位符。CPC 坐标定位符是距离在机器人装置 1 中设置的参考位置的坐标，并由一般的连接坐标系统表示，由用于坐标变换的旋转矩阵和位置矢量的垂直连接构成。

此外，所述标记指明相应的对象、所述对象的加载和为建立对象对对象的通信所需的数据。还将应用和设计标记相比作为更一般地更广泛地描述的所述虚拟标记以及设计标记与虚拟标记的合成标记。虚拟标记在由设计机器人装置利用设计标记替换之后被使用。

下面的公式表示设计标记的描述。设计标记由一个以上的 CPC 基本位置信息或设计标记构成。

Design Label(设计标记)

CPC Primitive Location Information(基本位置信息)or Design Label(或设计标记);

.....

) .....(1)

在另一方面，下面的公式表示虚拟标记的描述。虚拟标记由设计标记、CPC 基本(primitive)信息和 CPC 基本信息的 CPC 坐标定位符构成。

Virtual Label(虚拟标记)

Design Label(设计标记);

CPC Primitive(基本信息);

CPC Coordinate Locator(坐标定位符);

) (2)

以下的公式表示合成标记的描述：

Composite Label(合成标记)

Composite Label(合成标记)：

or/and Design Label(或/与设计标记);

(or/and Virtnal Label(或/与虚拟标记)：

) (3)

因而，在被所通知的CPC连接信息规定的CPC基本位置信息中，在被规定的CPC坐标定位符的树结构的顶边侧具有在虚拟标记中定义的CPC坐标定位符的情况下，设计机器人装置利用设计标记替换虚拟标记，此外，设置这样定义的设计标记为有效。

按照本实施例，通过使用CPC连接信息访问这种设计文件，可以确定当前机器人装置的配置(四腿型，轮型等)。此外，可以获得为更新配置依赖软件所需的各种数据。

图7是详细表示这种设计机器人装置的处理步骤的流程图。当CPC连接信息从虚拟机器人装置输出时，设计机器人装置从步骤SP1移动到步骤

SP2，并使用 CPC 连接信息访问设计文件，确定机器人装置 1 的当前配置。

然后，设计机器人装置被移动到步骤 SP3，比较由从虚拟机器人装置对象输出的 CPC 连接信息提供的 CPC 基本位置信息和由存储在设计文件中的 CPC 连接信息提供的 CPC 基本位置信息。

接着，设计机器人装置进入步骤 SP4，根据比较结果访问设计文件，相应于当前配置和用于重构在对象之间进行通信所需的数据，检测用于指定对象的设计标记。

然后，设计机器人装置移动到步骤 SP5，访问检测到的设计标记中的连接文件，并通过检测相应于所述标记的连接数据，获得为指定相应的对象和为重构对象之间的通信所需的数据。在这种结构中，连接文件是这样一种文件，其中相应于该标记记录有连接数据，所述连接数据是为指定对象和为重构对象之间的通信所需的数据。

接着，设计机器人装置移动到步骤 SP6，向对象管理器发送所述连接数据，并因而指定配置依赖软件的更新，移动到步骤 SP7，终止该处理过程。

图 8 表示设计文件的一部分。在所述设计文件中，DRX 表示这个机器人装置 1 的配置，4Legged 表示四腿型的配置。此外，Head,RightFrontLeg,LefrontLeg,RightRearLeg,LeftRearLeg 分别表示头，右前腿，左前腿，右后腿，左后腿。这样，在第一段中便定义了由头，右前腿，左前腿，右后腿和左后腿构成的四腿型。

此外，的第二段中 Wheel 表示轮子型由头，右前腿，左前腿，右后轮，左后轮构成，其和第一段不同。在第三段中，描述了摄像头连接到头的设置。在第四段中，描述了这一镜头连接的虚拟标记，在第五段以及后面的部分中，以树格式描述了关于右前腿，左前腿等的 CPC 坐标定位符。

在另一方面，图 9 表示 CPC 连接信息的描述的一部分。在本实施例中，CPC 连接信息由和设计文件的格式类似的文本数据格式描述。在这种描述中，关节表示接合部，并表示该部分是和其它部件的连接点。在这种情况下，显然，由第二行到第五行的描述可知，电视摄像头通过 5 个枢轴 CO，CO，CO，CO，CO 被连接。

因而，按照图 9 的例子的 CPC 连接信息和上面图 8 所示的四腿型的一致。

在另一方面，接着是一个表示连接文件的描述的一部分的图。连接文件

以文本数据格式进行描述，并且在第一段和第二段中，分别描述了相应于四腿型和轮胎型的腿和轮胎的对象，并且还要描述为在对象中构建对象之间的通信所需的数据。

更具体地说，在第一段中，描述作为服务名的对象名 MoNet 和数据格式 MoNeTOutData 和表示主体的 S。此外，作为相同的服务名，例如对象名 MoNetReplay，描述了数据格式 MoNetOutData。最后，设置表示观察者 O 的描述。此外，在下面的行中，相同的主体和观察者的诸如数据格式将被描述。

```
DRX 4legged (
    MoNet.MoNetOutData..S,MoNetReplay.MoNetOutData...0
    Monet.MoNetOutData..S,WalkingPatternGenerator.MoNetOutData...0
)
DRX Wheel (
    MotionConverter.MCData..S,MotionReplay2.MCData...0
    MotionConverter.MCData..S,Whell.MCData..0
)
```

…(4)

这样，在图 8 所示的例子的情况下，设计机器人装置从第一个语句的描述中按照这个 DRX 4Legged(带腿)提取连接数据(在 DRX 4Legged 之后的段中所述的)，并向对象管理器通知所述数据。因而，在本实施例中，将更新配置依赖软件。

图 10 是表示在设计机器人装置中在引导时的处理的定时图。在机器人装置 1 中，当电源被输入或被复位时，引导处理开始。此时，每个对象从文件中被加载，并在机器人装置 1 中产生配置依赖软件。并且此时每个对象管理器的内部变量被初始化。

然后，从对象管理器向每个对象传递 DoInit，并按照这个 DoInit，在对象管理器中记录每个对象的请求入口。这样，可以在机器人装置系统软件层中建立在对象之间的通信。

此后，DoStart 被从对象管理器按照顺序传递给虚拟机器人装置和设计机器人装置，因而虚拟机器人装置和设计机器人装置开始操作。利用这种结

构，由虚拟机器人装置产生CPC连接信息，并且这一CPC连接信息通过设计机器人装置的请求被输入到设计机器人装置，并且形成连接数据。

此外，这一连接数据被输送到对象管理器，并且利用所述的连接数据构成配置依赖软件。

在另一方面，在停止时，从对象管理器向每个对象传递DoStop,对于所有的请求都产生错误，因而在对象中的操作被停止。然后，从对象管理器向每个对象传递DoDestroy，每个对象释放使用的资源，并撤销在对象管理器中记录的入口，因而处于备用状态。

这样，即使在改变配置使系统升级的情况下，机器人装置1也可以使用适用于改变的配置的软件进行操作。

在另一方面，在插入和拔出的操作中，如图11所示，从对象管理器向每个对象传递DoStop，因而在每个对象中，通过所述DoStop对于所有请求都产生错误，因而停止操作。

然后，如同在引导的情况下那样，从对象管理器向虚拟机器人装置和设计机器人装置相继地传递DoStart，因而由虚拟机器人装置产生CPC连接信息。然后，所述CPC连接信息被传递给设计机器人装置，并产生连接数据，因而由所述连接数据形成配置依赖软件。

因而，在机器人装置1中，即使在通过插入和拔出动态地改变配置的情况下，机器人装置1也可以利用适用于改变后的配置的软件进行操作。

另外，当电池17被用光时，则由来自电池管理器的状态改变的请求重复执行DoStop和DoStart的处理，如同在插入与拔出的情况一样。因而，在电池17被用光时，状态将被改变，以便减少时钟频率并停止不需要的设备操作，使得能够通过钮扣电池29进行操作。在另一方面，在提供有充电电池17的情况下，将使时钟频率增加并启动每个装置的操作，并且由电池17进行操作。

(1-4)对象管理器的处理

对象管理器根据来自设计机器人装置的通知重构包括配置依赖软件的对象，因而其更新配置依赖软件。更具体地说，对象管理器根据来自设计机器人装置的通知加载和卸载对象，并且同时按照所述的加载和卸载重构对象之间的通信，这样，对象管理器便改变配置依赖软件。

如果在构造对象之间的通信的情况下相应的对象名在对象中被记录，则

每个对象的独立性将被破坏，因而需要准备多种对象以便和自由配置相适应。因而，按照本实施例，对象管理器根据要被从设计机器人装置发送的连接数据构造对象之间的通信，因此将保证对象的独立性。

在下面的说明中，描述将被简化，不指出在机器人装置 1 中的详细对象名。此外，将同步地执行对象之间的通信。

更具体地说，图 12 表示涉及本实施例的在两个对象之间的对象对对象通信的基本结构。按照本实施例，使用 Ready 和 Notify 执行对象对对象的通信，并且超过对象的处理能力的数据(在本例中，对象 B)将不被发出。

更具体地说，在图 12 中，对象 A 从作为对象 A 的成员的类 0 主体向作为对象 B 的成员的 0 观察者类发送数据，因而，将启用对象 B 的方法。

按照这种关系，只有当 Ready(准备好)信号被从观察者发出时主体才发送数据。Ready 信号应当每一个数据发出一次。这样，观察者侧的对象 B 在等待完成收到的数据的处理时可以接收后继的数据。

在另一方面，图 13 表示在多个观察者中在对象之间通信的基本结构。在这种情况下，作为主体的对象 A 可以给所有的观察者分配数据，或者只对由对象 ID 指定的特定的观察者分配数据。按照本实施例，即使在这种多观察者的情况下，也可以通过 Ready 信号把数据发送给相应的观察者。

在每个对象中，通过指定用于规定对象的对象 ID 和用于定义方法的选择器号(方法号)，相应的方法将被启动(入口)，并且指定观察者，因而能够发送所需的数据。

图 14 是表示在多主体中对象之间通信的基本结构的示意图。在这种情况下，主体 A 和 B 是主体，对象 C 是观察者。观察者可以分别从多个主体接收数据，并且在每当其接收数据时，将启用用于处理该数据的方法。此外，观察者可以通过只对由主体 ID 指定的特定主体输出 Ready 信号来选择性地输入所需的数据。

按照本实施例，在至少该配置依赖软件属于它的主体中，如参照图 12-14 所述，将执行使用 Ready 和 Notify 在对象之间进行通信。此外，当偶尔需要时，可以按照多观察者和多主体的连接模式连接其它的对象。更精确地说，具有多个观察者的对象也具有相应于多个观察者的多个连接入口。

图 15 表示对象管理器和每个对象之间的关系。在每个对象中，由对象 ID 和选择器号指定方法，并改变数据，并且选择器号 0-3 将被分配给基本方

法，而和相应的主体和观察者无关。

其中 DoInit 被分配给 0 号选择器，当该 DoInit 开始时每个对象被初始化。此外 DoStart 被分配给 1 号选择器，当所述 DoStart 开始时，每个对象开始操作。此外，DoStop 被分配给 2 号选择器，当 DoStop 开始时，每个对象停止操作。此外，DoDestroy 被分配给所述 2 号选择器，按照所述 DoDestroy，每个对象释放资源。

此外，用返回值，响应于来自对象管理器和其它对象的请求，每个观察者通知所述观察者的服务名。

因而，按照本实施例，根据被设计机器人装置通知的连接数据，在由对象管理器加载和卸载之后，可以重构对象之间的通信。

下面的公式表示要从设计机器人装置输出的连接数据，所述连接数据以文本数据格式描述，由冒号分开主体的服务名和观察者的服务名。在这种情况下，在主体侧，主体名 FooB,数据格式 Image,由相应的部件名 RighCamera 确定的名称，表示主体的 S，作为服务名被描述了。此外，在观察者侧，主体名 FooA,数据格式 Image,由相应的部件名 RighCamera 确定的名，表示观察者的 O，作为服务名被描述了。

FooB.Image.RightCamera.S.FooA.Image.RightCamera.0 .....(5)

在这种情况下，当加载对象时，对象管理器根据所述连接数据的描述检测要被加载的对象名。此外，对象管理器确保获得堆栈(stack)存储器和堆(heap)存储器，并从预定的文件加载具有所述检测的对象名的对象。此时，对象管理器从操作系统 OS 获得对象 ID，并和在连接数据中描述的对象名一道记录所述对象 ID。

因此，对象管理器利用如此登记的对象 ID 进行诸如 DoInit, DoConnect; Dostart 的处理。

更具体地说，在 DoInitialize 中，对象管理器把 DoInit 调用到通过根据获得的对象 ID 和上述的选择器号加载对象而获得的所有的对象中。每个对象按照所述 DoInit 初始化内部变量，因而，对象管理器初始化每个对象。

在此初始化处理中，按照每个对象的通知，对象管理器登记每个对象的作为主体(控制)的入口和作为对象(连接)的入口。所述登记由对象名和相应于

在连接数据中的主体和对象的描述的主体和对象名构成。

在另一方面，在 DoConnect 中，对象管理器根据在 DoInitalize 中的登记向具有观察者的每个对象的入口(连接)通知主体的主体 ID(对象 ID)和入口(控制)。对象在接收这一信息时，对象便使用所通知的主体 ID 和入口(控制)启用相应的主体，并且入口(通知)被连接和登记。此外，Ready 从向其请求连接和登记的主体返回观察者。因而，观察者和相应的主体被连接。

在重构所述对象和所述主体之间对象对对象的通信的情况下，对象管理器将主体 ID(对象 ID)和入口(控制)告知由连接数据告知的观察者。因而，按照本实施例，每个对象可以开发出来，而不用清楚地表示要被连接的对象，并且在偶而需要时，可以通过对象管理器的指令和各个对象进行连接。因此，可以获得与过去相比更高的独立性。

此外，此时，通过按照连接数据由对象管理器构成对象对对象的通信，可以简单而自由地更新配置依赖软件，因而可以简单地获得适用于该配置的软件。

在另一方面，在 DoStart 中，对象管理器通过上述的 1 号选择器向每个对象命令 DoStart。每个对象，如果具有观察者，就使用由 DoConnect 获得的主体 ID 和入口 Ready 向主体发送 Ready。因而，相应的主体形成数据可接收状态，并且配置依赖软件启动所述操作。

更具体地说，包括多个观察者的主体向被登记的观察者中的正输出 Ready 的观察者通知例如作为传感器输出的数据。此外，观察者在完成接收的数据的处理时，发送 Ready，使得可以接收后续的数据。

在另一方面，在关闭(Shut Down)、插入(plugIn)与拔出(plugout)以及上述的状态改变的情况下，因为从设计机器人装置通知的连接数据与设计机器人装置发送出的预先登记的内容不同，所以对象管理器使用 2 号选择器向每个对象发出 DoStop。在这种情况下，观察者就取消入口 Ready。

此外，在 DoDisconnect 中，对象管理器中断主体和观察者之间的通信。在这种情况下，对象管理器通过向观察者(连接)的入口发送 DoDisconnect 消息，使观察者向该观察者中的相应的主体发出入口(控制)的断开请求(除去观察者)，因而通信连接便被断开。

在另一方面，在 DoDestroy 中，对象管理器通过使用上述的 3 号选择器向相应的对象发送 DoDestroy，因而，对象将被破坏。此时，在这一对象中，

将执行在 DoInit 时执行的登记的取消。

在对象卸载时，对象管理器释放在被 DoDestroy 破坏的对象上的堆栈存储器和堆存储器的区域，并卸载这个对象。此外，对象管理器破坏在加载时登记的主体 ID 和主体名。

当对象管理器通过这些控制从设计机器人装置收到例如连接数据时，它便按照图 16 所示的顺序进行控制，直到 DoStart。

更具体地说，当由消息发送连接数据时，对象管理器启动，并加载例如在所述连接数据中所述的对象 A 和对象 B。这里，对象管理器按照操作系统的系统命令加载对象 A 和对象 B。此时，对象管理器获得对象 A 和 B 的对象 ID，并记录这些 ID。

接着，对象管理器利用上述获得的观察者 ID 和 0 号选择器启动对象 A 和 B 的 DoInit。并因而其从对象 A 和 B 获得作为主体(控制)的入口和作为对象(连接)的入口，并记录它们。

然后，对象管理器通过使用被登记的入口(连接)启动作为观察者的对象 A 的 DoConnect，并连接作为主体的对象 B，以及对象 A。因而，根据连接数据构造对象对对象的通信，并且对象管理器启动对象 A 和 B 的 DoStart。

图 17 是表示在 DoStart 之后的步骤的定时图。对象 A，对象 B 和其它对象通过对象管理器的 DoStart 启动，通过使用在这些对象中的所述 Ready 和 Notify 进行对象对对象的通信。

更具体地说，在这种情况下，当启动时，对象 A 向对象 B 的 Ready 入口发送 Ready 信息，因而由对象 B 通过对象 A 的 Notify 入口，数据被从对象 B 发送给对象 A。此外，如果在处理这个数据的期间没有从对象 A 向对象 B 的 Ready 入口发送所述消息，则登记用对象 B 的对象 A 的 Notify 入口。并且数据将由用对象 A 的对象 B 的 Ready 入口发出。因而，在对象对对象的通信中，可以防止超过对象 A 的数据处理能力的数据发送。

图 18 表示插入和拔出以及状态改变时程序的定时图。如果要从设计机器人装置输出的连接数据和以前登记的由设计机器人装置发送的内容不同，则对象管理器向所有对象发送 DoStop，并停止所有对象的操作。在这种情况下，在每个对象 A 和 B 中，通过取消入口 Ready，其自身的通知入口以后将不再启动。

当所有对象的 DoStop 完成时，对象管理器向对象 A 发出 DoDisconnect，

并在把对象 A 和其它对象分离之后，启动对象 A 的 DoDestroy。因而，对象 A 的资源将被释放，并且取消由 DoInit 执行的对象 B 的登记。

当对于所需的对象的 DoDestroy 完成之后，对象管理器按照操作系统的系统指令进行卸载。此时，调用所需对象的破坏程序，在这种情况下，对象 A 的破坏程序被调用，并取消在加载时执行的对象 A 的记录。此外，对象管理器释放栈存储器和堆存储器，因而，完成对象 A 的卸载。

然后，对象管理器按照连接数据命令加载对象 C，并在需要时以和上面图 16 所述的方式相同的方式启动 DoInit，DoConnect 和 DoStart。因而由对象 A 和对象 B 构成的结构被改变为在操作状态下不被编译的由对象 B 和对象 C 构成的结构。

(1-5)按照相关的存储器的对象的记录(registration)

图 19 按照式(5)所示的连接数据表示对象 A 和对象 B。对象管理器在相关的存储器中存储并记录所述连接数据，在启动对象的 DoInit 并记录每个对象时，根据连接数据的描述在相关的存储器中记录每个对象。

更具体地说，在这种情况下，在观察者侧对象 A 的对象是 FooA，在主体侧对象 B 的对象名是 FooB，数据格式是 Image，替换名是 RightCamera。因而，在该主体和对象的连接数据中，主体的服务名和观察者的服务名由冒号分开，并被表示为“FooB.Image.RightCamera.S.FooA.Image.RghtCamera.0”

当连接数据从设计机器人装置输出时，如图 20 所示，对象管理器记忆并存储主体侧用于描述连接数据“0.FooA.Image.RightCamera”的描述“FooB.Image.RightCamera.S”作为密钥。

在另一方面，每个对象通过 DoInit 响应于对象管理器的记录入口而通知服务名和选择器号，并且如上述的情况一样，对象管理器在相关的存储器 2 中存储这些通知，并记录每个对象。

更具体地说，在观察者侧的对象 A 通知服务名“FooB.Image.RightCamera.0”，其是和在连接数据中的观察者侧的描述相同，并且还通知连接入口的选择器号。在这一连接中，假定在所述对象 A 中的选择器号是 5。

如图 21 所示，对象管理器在相关的存储器 2 中记录对象 ID 和对象 A 的选择器号(OID.Entry)=(2,5),利用在连接数据中观察者侧的“FooA.Image.RightCamera.0”作为密钥。在这种连接中，对象 A 的对象 ID

作为值 1 由操作系统通知。

此外，在这种情况下，主体侧的对象 B 按照和主体侧“FooB.Image.RightCamera.S”的描述相同的描述通知服务名，并且还通知连接入口的选择器号。在这种连接中，在对象 B 中的选择器号将是 5。

如图 21 所示，利用对象 A 的描述，对象管理器在相关存储器 2 中记录所述对象 B 的对象 ID 和所述对象 B 的选择器号(OID.Entry)=(1,5), 利用主体侧的(FooB.Image.RightCamera.S)作为密钥。在这种连接中，对象 B 的对象 ID 作为值 2 由操作系统通知。

当进行 DoConnect 时，对象管理器从连接数据中分离主体侧的描述和对象侧的描述。因而，对象管理器分离连接数据的描述，以便和存储在相关存储器 1 中的密钥(FooA.Image.RightCamera.0)以及利用这一密钥(FooA.Image.RightCamera.0)在相关存储器 1 中存储的各个描述(FooA.Image.RightCamera.0)中的一个一致。

此外，对象管理器分别按照这些被分开的描述检索相关存储器 2，并检测相应的观察者的对象 ID 和选择器号(1，5)，以及主体 ID 和主体的选择器号(2，5)。

这个对象管理器向对象 ID 和选择器号(1，5)发送对象 ID 和选择器号(2，5)的消息，并启动观察者的连接入口，并通过对象 ID 和选择器号(2，5)给这个连接入口通知主体信息。此外，对象管理器利用来自这一连接入口的对象 ID 和选择器号(2，5)启动主体侧的控制入口。并把观察者侧(这种情况下是 Notify)的对象 ID 和选择器号(1, 6)记录在控制入口。此时，观察者侧通过返回值获得 Ready 的选择器号，并对此进行记录。

因而，对象管理器通过使用相关存储器 1 和 2 建立了对象之间的连接。

利用这种结构，当建立对象之间的连接时，对象管理器在相关存储器 3 中存储连接数据的主体侧描述(FooB.Image.RightCamera.S)，利用连接侧的服务器侧的描述(FooA.Image.RightCamera.0)作为密钥。此外，其从相关存储器的描述中取消了利用观察者侧的描述“FooA.Image.RightCamera.0”作为密钥的主体侧的描述“FooB.Image.RightCamera.S”。

因而，对象管理器在相关存储器 3 中按照相应于连接数据的描述记录并存储在对象之间的连接建立之后形成的观察者和主体。

在另一方面，在进行 DoDisconnect 时，对象管理器断开连接，和

DoConnect 时相反，并消除在相关存储器 3 中记录的相应的描述，在相关存储器 1 中对其进行重新记录，并取消在相关存储器 2 中的描述。

在另一方面，假定被下式表示的连接数据由于配置依赖软件的改变而由设计机器人装置通知。

FooB.Image.RightCamera.S.FooC.Image.RightCamera.0 (6)

这个连接数据是对于改变观察者到对象 C 代替对象 B 的描述。在这种情况下，对象管理器从相关存储器 2 中利用“FooA.Image.RightCamera.0”作为密钥检测对观察者设置的对象 ID 和选择器号(1，5)。此外，对象管理器，以及通过这一对象 ID 和选择器号(1，5)命令观察者 A 执行 DoDiSconnect,消除了相关存储器 2 的相应的描述(作为观察者和主体的对象 A 和 B 的描述)。并且对象管理器在消除被记录在相关存储器 2 中的描述之后，重新在相关存储器 1 中记录所述描述。

更具体地说，对象管理器从相关存储器 2 中消除利用 FooA.Image.RightCamera.0，FooA.Image.RightCamera.S 作为密钥形成的对象 ID 和选择器号的描述。利用这种结构，对象管理器可以动态地改变对象。

(2)第一实施例的操作

按照上述的结构，在机器人装置 1(图 1)中，如果腿作为可动单元 3 被连附于主单元 2 上，便成为四腿配置，并且可以利用被连附于这些腿上的电机驱动的四腿行走。此外，如果连附轮胎代替后腿，则成为轮子型配置，并可以利用这些轮子运动。

在机器人装置 1(图 2)中，所需的控制指令通过中央处理单元 22 的处理产生，并且运动单元 3 根据由腿设置的传感器例如机器人装置部件 24，25 和 26 的输出以及由被设置在头单元 2A 中的各种信息获取装置获得的信息被控制。

在机器人装置 1(图 3 和图 4)中，这些控制将通过使用上层的面向对象软件借助于驱动设备驱动器的各种设备驱动器执行。在设备驱动器的上层的机器人装置系统软件中，和串行总线相连的每个部件都要被虚拟机器人装置检测。并且表示在整个的机器人装置 1 中哪些机器人装置部件、是按照怎样的顺序连接的 CPC 连接信息被通知设计机器人装置。这样机器人装置 1 的当前配置便由和总线相连的装置确定，此外，可以精确地确定形成当前配置的要被控制的对象。

在机器人装置1中，这个机器人装置系统软件的上层的软件将被更新，这样，整个操作将被与该配置相应的合适的软件进行控制。

在更新软件时，机器人装置1比较通过给每个部件的CPC基本位置组附加标记而形成的设计文件和用机器人装置系统软件的设计机器人装置在每个配置中要由虚拟机器人装置形成的CPC连接信息，并选择适合于所述配置的合适的对象。然后按照这个对象配置依赖软件将被更新(图7和图8)。因而，在机器人装置1中，软件可以被简单地改变为适合于配置的合适的软件，使得可以控制整个操作。

此外，上层软件被分成不依赖于机器人装置的配置的配置独立软件和配置依赖软件。此外，在要由配置独立软件按照不依赖于配置的格式处理的数据被配置依赖软件转换为相应于配置的格式并在机器人装置系统软件之间接收与/或发送的条件下，只有所述的配置依赖软件被更新。因而，软件可以通过简单的处理被更新。

此外，此时，在虚拟机器人装置中，通过文本数据格式的描述形成CPC基本位置信息(图8和图9)。而在设计文件中，因为CPC基本位置信息按照相同的文本数据格式描述，所以即使要被改变的配置在将来被增加的情况下，也可以容易地处理。

在机器人装置1中，由对象管理器按照配置依赖软件的各对象的重构来执行软件更新的处理。更具体地说，当启动时，指定适用于当前配置的对象连接的连接数据从设计机器人装置输出到对象管理器，并且要被这一连接数据指定的对象被加载(图16)。

此外，在被加载的对象初始化之后，通过在对象管理器指定连接数据使主体和对象连接，使得可以交换数据。因而，在机器人装置1中，配置依赖软件将由适合于该配置的对象构成。

因而，在机器人装置1中，每个对象在设计时可以不考虑要被连接的具体对象，并且可以构成软件。此外，因为连接数据可以按照配置被形成，所以可以简单地设计适合于配置的软件。此外，不进行重新编译和重新链接的处理便可以改变软件。因而，在机器人装置1中，和常规的装置相比，对象的独立性被显著地增加，可以简单而快捷地提供适用于自由配置改变的软件。此外，可以通过设置连接数据在一个对象和多个对象之间构成对象对对象的通信。

因为这个连接数据是用于指定和主体通信的数据格式的数据，所以通过指定相应的对象 ID 和与所述连接数据相应的选择器号，对象管理器可以构成对象对对象的通信。

此外，设计机器人装置通过从记录有多个连接数据的设计文件中选择连接数据并提供给对象管理器，在偶而需要时通过这个设计设置可以构成和各种配置兼容的软件。

此外，因为这个连接数据以文本数据格式描述，使得可以简单地改变为各种格式。

在另一方面，当配置被改变时，由机器人装置 1 中的虚拟机器人装置检测配置的改变。并把有关所述被改变的配置的 CPC 连接信息通知设计机器人装置。此外，在这种设计机器人装置中，根据与设计文件的比较形成和被改变的配置相关的连接信息。此外，按照这个连接数据，由于配置的改变而不再需要的对象由对象管理器卸载，由于配置的改变而所需的对象被加载，因而将重构包括配置依赖软件的对象。此外，这些对象将被这样连接，使得可以按照连接数据进行对象对对象之间的通信，这样，配置依赖软件将被更新。

利用上述的结构，在机器人装置 1 中，在需要时可以更新配置依赖软件，可以在配置独立软件处于操作状态的情况下构成适合于配置的软件。此外，可以对这种软件改变加上供电方式的改变。

(3)第一实施例的效果

按照上述结构，通过比较作为部件检测结果的连接数据和由相应于每种配置的部件检测结果的一组数据构成的设计文件，并且根据所述比较结果按照配置改变软件，可以按照伴随着部件改变的配置的改变来改变软件。因而，可以简单地设置相应于当前配置的合适的软件。

此时，通过按照对象的改变来改变软件，可以用简单的处理改变软件。

此外，因为上级软件由配置依赖软件和配置独立软件构成，并相应于配置通过改变配置依赖软件改变软件，所以可以简单地提供适合于配置的软件。

此外，因为相应于部件检测结果的数据利用文本数据描述，所以即使在要被改变的配置在将来增加的情况下，也可以容易地处理。

此外，按照上述结构，在其部件被交换因而可以改变配置的机器人装置

中，对象管理器连接各对象，使得可以进行数据交换。因而，和过去相比，可以改善对象的独立性，因而可以容易地改变软件使得适合于所述配置。此外，软件可被简单地改变而不进行重新编译和链接。此外，通过设置连接数据，可以在一个对象和多个对象之间构成对象对对象的通信。

此时，因为连接数据是用于指定对象和用于通信的数据格式的信息，所以这些对象可以容易地被连接。

此外，因为连接数据从记录有多个连接数据的并且各对象被连接的设计文件中选择，所以通过设置这个设计文件可以简单地提供适合于配置的软件。

此外，按照连接数据，通过在分离相应的对象之间的连接之后连接各对象来构成对象对对象的通信，可以按照需要更新软件，因而可以在软件正在运行的情况下构成和配置相应的软件。

此外，通过由虚拟机器人装置检测部件并提供连接数据，可以相应于配置改变软件。

此外，因为连接由文本数据描述，所以可以简单而快捷地构成相应于配置的合适的软件。

此外，通过使用 Ready 和 Notify 在对象当中交换数据以便进行通信，所以可以避免超过处理能力的数据供应。

(4)第二实施例

在由 Call 指令处理例如上述的 DoInit 指令的操作系统中，通过输出该指令把处理移交给输出目的地。因而，对象管理器中断从每个对象发送的信息，直到从目的地返回 Reply。因而，如图 16 所示，在发送 DoInit 之后在对象管理器记录作为主体(控制)的入口和作为对象(连接)的入口是困难的。

因此，在按照本实施例的机器人装置中，如图 22 所示，和图 4 相比，这些入口(控制，连接)被记录在服务管理器中，并且服务管理器控制对象对对象的通信，在另一方面，对象管理器只执行例如对象的加载和卸载的处理。更具体地说，按照本实施例，对象管理器命令停止操作，释放用于对象对对象的通信的连接，从而破坏对象，释放构成配置依赖软件的所有对象上的资源。此外，对象管理器命令新需要的对象的加载和初始化，并指令对象的启动。在另一方面，服务管理器控制在对象对对象的通信中的通信方的信息。

因为除去对象管理器和服务管理器的结构以及和这些结构相关的每个部件之外，本实施例和第一实施例相同，因而下面着重说明对象管理器和服务管理器的结构，并省略相同的部分。

按照图 23 所示的实施例，对象之间的连接意味着用于在对象之间进行接收/发送数据的连接，并且被表示为进行通信的双方服务之间的连接。此外，如式(5)所表示的，所述连接被表示为数据输出侧的服务，数据输入侧的服务，和通信数据的类型，以及数据流的方向。此外，如图 24 所示，连接状态意味着在机器人系统中加载的对象组中的对象当中建立的各种连接的组合。并且连接数据意味着当在开始时配置被改变时在对象当中新建立的连接数据，它是当前的连接状态和新的连接状态之间的差值数据。此外，断开数据意味着当配置被改变时该连接数据成为不需要的，并且是新的连接状态和当前的连接状态之间的差值数据。

图 25 是和图 10 对照的关于本实施例的在启动机器人装置系统软件时的定时图。当电源输入或复位时，首先对构成机器人系统软件的每个对象执行初始化处理。更具体地说，在机器人装置中，如图 26 所示，从对象管理器到电源管理器、服务管理器、虚拟机器人装置和设计机器人装置按照所述顺序重复进行 DoInit 的发送和 Reply(应答)的接收。因而，在机器人系统软件中将执行 DoInitialize 阶段处理。在下面一节的说明中，处理顺序用图中的标号表示。

此外如图 27 所示 DoStart 按照下面出现的顺序连续地发送给到电源管理器、服务管理器、虚拟机器人装置和设计机器人装置。因而，在机器人系统软件中执行 DoStart 阶段的处理，所述电源管理器、服务管理器等开始操作。

此时，设计机器人装置通过发出 DoStart 命令向虚拟机器人装置输出表示机器人装置的当前配置的信息。并且虚拟机器人装置向设计机器人装置通知表示机器人装置部件以何种顺序连接的 CPC 连接信息。设计机器人装置从虚拟机器人装置接收这个 CPC 连接信息，并在由所述 CPC 连接信息形成连接数据之后向对象管理器返回 Reply。

因而，当执行 DoStart 阶段时，由机器人装置中的对象管理器向服务管理器发出 DisConnect(断开)指令。在启动时，因为没有对象被加载，所以服务管理器按照这个命令从设计机器人装置获得连接数据，并向对象管理器通知处理的完成。并且类似地，对象管理器从设计机器人装置获得连接数据。

因而服务管理器获得用于在构成配置依赖软件的每个对象上的各对象间进行通信所需的信息。在另一方面，对象管理器获得用于加载和卸载这些对象所需的信息。

因而，对象管理器形成相应于当前配置的必需的对象和不需要的对象的表。此外，如图 28 所示，对象管理器指定这些需要的对象 A-C 加载，因而安装这些对象 A-C。此时，对象管理器通过执行所述加载指令获得每个对象的对象 ID(OID-a,OID-b,OID-c)，如图 29 所示。

然后，如图 30 和 31 所示，对象管理器通过这样获得的对象 ID 对每个对象连续调用 DoInit。每个对象通过该 DoInit 请求把每个服务记录到服务管理器。其中，通过记录每个服务名、对象 ID 和选择器执行所述的服务记录。

当服务管理器响应每个对象的请求记录每个对象的服务时，则向每个对象通知记录完成。并且响应这个通知，每个对象向对象管理器返回 Reply。

当按照上述安排完成每个对象的初始化时，对象管理器命令每个对象执行对于服务管理的 DisConnect。此时，如图 32 所示，服务管理器，通过获得设计机器人装置在 DoStart 阶段获得的连接数据，形成一个表并存储这个表。这个表包括和 Connect 有关的表、和 DisConnect 有关的表，并且每个表包括每个对象的服务名和按照上述第一实施例的连接数据的数据格式。此外，服务管理器在服务表中保持通过每个对象的 DoInit 而获得的服务。其中，其中，这个服务表被编译并在每个对象中形成服务信息，并且每个服务包括和每个服务相关的对象 ID，用选择器的服务入口，和每个服务名。

如图 33 所示，当对象管理器，命令执行对象的 Connect 时，服务管理器则参考从每个对象获得的服务数据和从设计机器人装置获得的连接数据。此外，根据这个参考结果，服务管理器按顺序记录要和每个对象通信的对象，并连续地连接每个对象，使得可以进行数据通信。

更具体地说，如图 34 所示，当服务从每个对象被记录时，服务管理器则形成具有每个服务的服务名的服务数据和在每个对象中服务入口的服务信息。因而，服务管理器形成一个服务表。并且，服务管理器通过从通过设计机器人装置获得的连接数据掌握新的连接状态和当前的连接状态，和在过去记录的连接数据，形成连接数据和断开数据。此外，服务管理器由连接数据和断开数据形成关于连接的表和关于断开的表。

服务管理器通过参考服务表和关于连接(Connect)的表，进行要被连接的

服务入口的检测，并根据检测结果，服务管理器向相应的对象通知该服务入口。此外，通过参考服务表和关于 DisConnect 的表，服务管理器检测服务入口，以按照这个检测结果断开连接，其向相应的对象通知该服务入口。在这方面，如图 30 所示，因为处理是在对象尚未被连接时的开始时的处理，所以服务管理器进行一系列的处理而不形成断开信息。这样，如图 35 所示，服务管理器连接该方法，使得可以在观察者和主体之间交换数据。图 36 是和图 19 对照的用于表示被这样连接的对象 A 和对象 B。

然后，当通过连接信息使连接完成时，服务管理器通知对象管理器连接完成(图 30)。当对象管理器获得这个通知时，命令每个对象进行 DoStart，如图 3B 所示。因而，如和图 28 对照的图 38 所示，每个对象 A-C 启动操作使得可以在连接之后进行数据通信。因而，在机器人装置中，可以根据对象之间的数据通信按照配置执行各种操作。

图 39 是和图 18 对照的表示由 PlugIn 和 PlugOut 改变配置时的程序的定时图。如图 40 所示，当机器人装置通过虚拟机器人装置改变配置而被构成时，则进行一系列的顺序。按照这一通知，对象管理器向位于机器人系统软件的上级的每个对象(图 39 的对象 A-C)连续地发送 DoStop 如图 41 所示，因而停止这些对象的操作，此外，对象管理器向设计机器人装置、虚拟机器人装置服务管理器和电源管理器以和启动时相反的顺序连续地发送 DoStop，因而控制这些对象的操作。其中，在构成机器人系统软件中的每个对象中，和上级对象有关的处理将被 DoStop 停止，而构成机器人系统软件的对象之间的处理例如电源管理将被继续。

利用这种结构，当在所有对象的 DoStop 处理被完成时，对象管理器向构成机器人装置系统软件的电源管理器、服务管理器、虚拟机器人装置和设计机器人装置按照启动时的顺序连续地发送 DoStart 指令。因而，对象管理器命令操作电源管理器、服务管理器等。

此时，设计机器人装置从虚拟机器人装置如同图 27 所述的情况那样获得 CPC 连接信息，并由 CPC 连接信息构成连接数据。在这种连接中，在对象管理器中，其认为在 DoStop 之后经过一个固定的时间间隔之后将命令 DoStart，使得由用户完成配置改变之后可以获得连接数据。

因而，当通过发送 DoStart 从电源管理器获得应答时，对象管理器命令服务管理器 DisConnect。在配置改变的情况下，因为当前加载的对象和相应

于配置改变的对象不同，所以服务管理器通过检查连接获得关于由设计机器人装置改变的配置的连接数据。此外，服务管理器从获得的连接数据形成断开数据，如上面图 34 所述，并按照这个断开数据解除相应对象的连接。

因而，当完成断开时，服务管理器通知对象管理器处理已完成。类似地，对象管理器从设计机器人装置获得连接数据，如图 44 所示。此外，对象管理器根据这样获得的连接数据对相应的对象调用 DODestroy，并释放由相应的对象占据的存储器。

此外，对象管理器命令卸载这个对象。服务管理器切断对象之间的连接，使得由于按照一系列处理而发生的配置改变变得不再需要的对象不再能够进行对象之间的通信，同时对象管理器卸载成为不需要的对象。

因而，当对象管理器卸载不需要的对象时，在命令新变成需要的对象加载之后，对象管理器命令所述被加载的对象 DoInit。接着，对象管理器命令服务管理器进行对象之间的连接。并且对象管理器通知连接数据，使得根据图 34 所说明的连接数据可以进行数据通信，如同在相应的对象之间开始时进行通信时那样。因而对象被这样连接，使得数据通信是可能的，对象管理器向每个对象发送 DoStart，并结束一系列处理。按照一系列处理，对象 D 代替对象 C 被记录如和图 37 对照的图 4 中的对象 A-D 的关系所示，因而对象 A，B，D 开始图 47 所示的操作。

图 48 是表示在开始时关于配置改变的时间的服务管理器的一系列处理的状态转换图。更具体地说，服务管理器，在被初始化之后，处于备用状态，并利用来自对象管理器的指令形成断开数据和连接数据，其按照连接数据连接每个对象。此外，当配置改变时，在由断开数据断开相应的对象的连接之后，服务管理器通过连接数据连接每个对象。

每个对象被这样连接，使得可以进行数据通信，其中把在主体侧的存储器区域中的本地数据复制到观察者侧的存储器中，或者使用在主体和观察者之间的公共存储器进行数据交换。

图 49 是表示通过复制进行数据通信的示意图。主体把在存储器区域中存在的本地数据复制到观察者侧的存储器区域中，并向在观察者侧要被调用的方法通知复制接收方的头部地址。因而，在机器人装置中，当数据量小因而通信的参与方少时，可以简单而容易地进行数据通信。

在另一方面，图 50 是表示利用公共存储器进行数据通信的示意图。主体

把数据写在存储器区域被映射的公共存储器中，并作为信息向观察者传递这个公共存储器的地址。利用这个输出的地址，观察者在把这个公共存储器区域映射到各个存储器区域之后，观察者使用公共存储器区域。因而，在机器人装置中，在发送大量数据例如图像数据和音频数据的情况下，可以应用于具有大量通信目的地的情况，并且可以容易而可靠地进行数据交换。

因而，在利用公共存储器进行通信的情况下，可以在对象中对接收数据的对象数量进行计数。更具体地说，如图 51 所示，在数据可以重写的对象中，当数据被写入公共存储器中时，相应的计数器的值被设置为 1。这个数据可重写的对象具有已接收的数据作为成员的主体。

主体使计数值增加 Notify 被执行的数，并且对象返回已在 Ready 情况中用完的区域中的 ID。当对象接收 Ready 时，相应的公共存储器计数器的值增加 1。

因而，这个数据可重写的对象可以按照这个计数值判断由其它的对象进行的公共存储器的访问是否已被完成，并根据判断结果，公共存储器可以被重写，因而可以可靠地保护数据。

因而，在通过复制或者通过公共存储器进行数据通信的情况下，如图 52 所示，在机器人装置中，被用于数据通信的数据利用预定的指令被连续地复制在观察者的存储器区域中，或者被写在公共存储器中。此时，这多个数据是被通知的地址信息，使得通过预定的命令可以在观察者获得这些数据。因而，在机器人装置中，多个数据可以立即被分配，因而可以改善可用性。

按照第二实施例，在通过 Call 指令进行处理例如 DoInit 的命令的操作系统中，可以简单地设置适合于当前配置的合适的软件，因而和现有的常规装置相比，可以改善对象的独立性。

(5)其它实施例

上述的实施例涉及预先设置选择器号例如 DoInit 的情况。不过，本发明不限于此，通过利用连接数据启动每个对象的 DoInitialize 也可以设置选择器号。此外，通过对对象管理器按照对象的记录启动 DoStar(primitive)

t，选择器号可以用不同的方式设置。

此外，上述的实施例涉及只通过配置依赖软件构成对象的情况。不过，本发明不限于此，可以广泛地应用于利用连接数据构成所有软件的情况，和通过连接数据只构成设备驱动器层的情况。

此外，上述的实施例涉及本发明应用于利用由观察者和主体构成的观察者模式的情况。不过，本发明不限于此，而是可以广泛地应用于例如服务器-客户机系统。在这种情况下，客户机接收连接数据作为数据输入，并使数据和服务器相连，从而使系统在构成时可以不需要用源代码描述的服务器名。因而，客户机和服务器分别构成对象。

此外，上述的实施例涉及通过 Ready, Notify 通信交换数据的情况。不过，本发明不限于此，而是可以广泛地应用于通过各种方法进行的对象之间的通信。

按照上述的本发明，通过根据比较在每个配置中保存的相应于部件检测结果的信息和关于当前配置的部件检测结果，按照配置改变软件，可以简单地设置适合于当前配置的软件。此外，根据连接数据，可以使对象被连接使得可以交换数据，因而，和过去相比，可以改善对象的独立性。

工业实用性

本发明可以用于宠物型机器人装置和服务器-客户机系统。

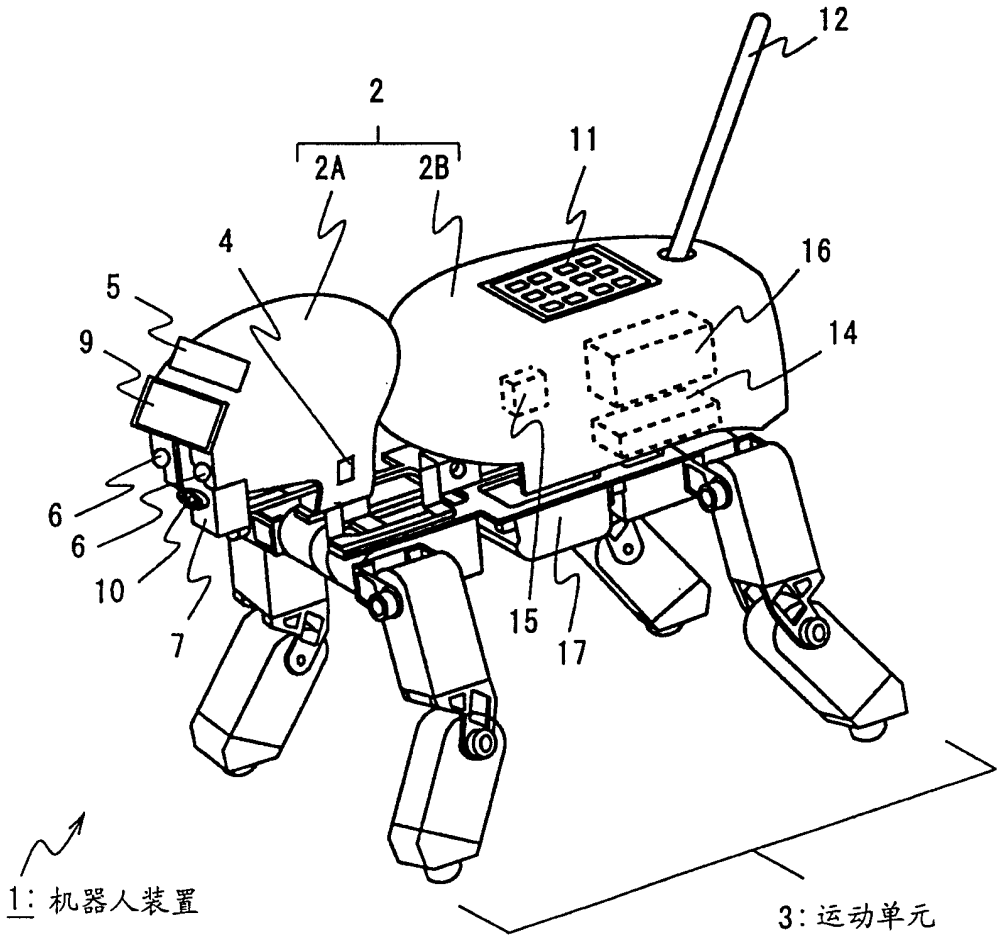


图 1

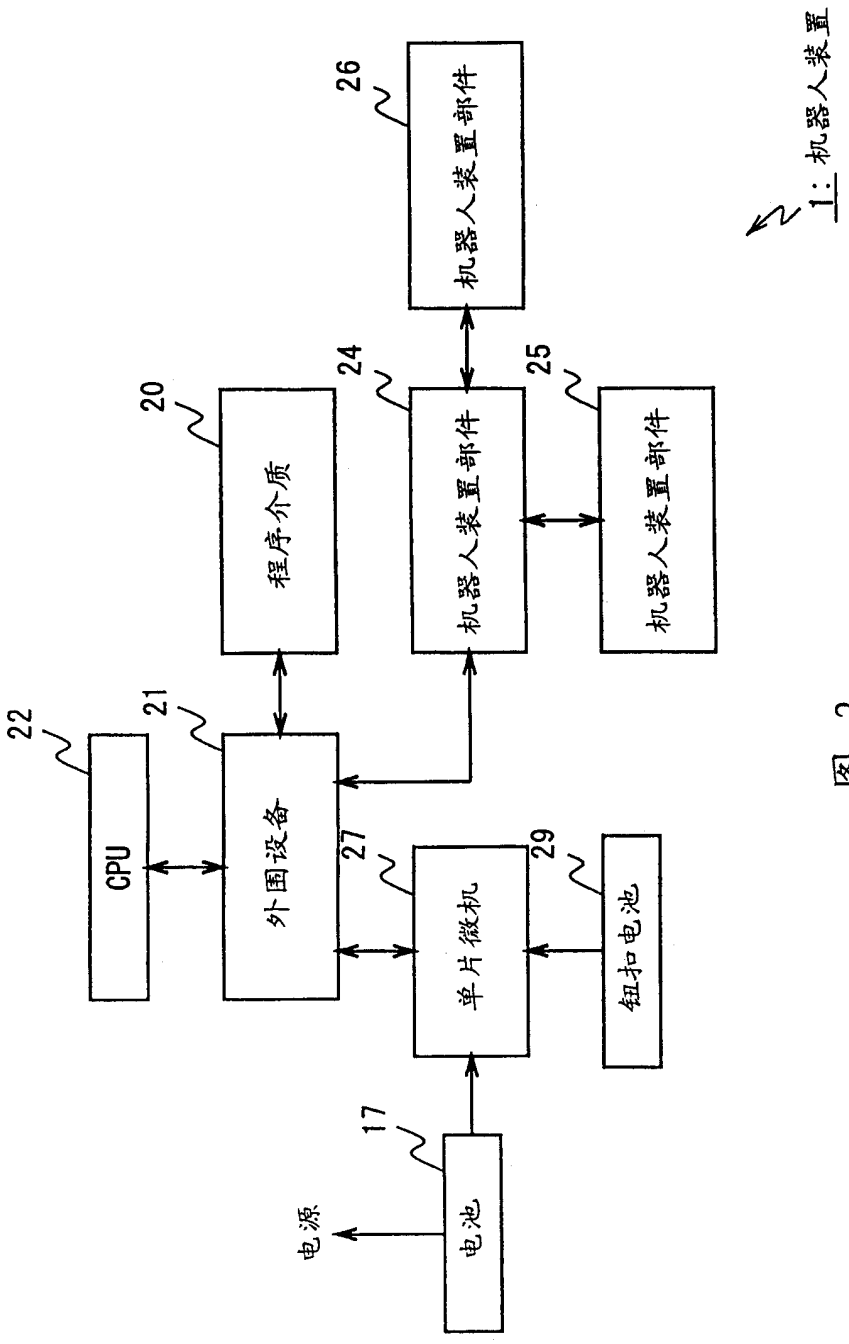

1：机器人装置
26
机器人装置部件
24
25
20
程序介质
机器人装置部件
机器人装置部件
22
21
CPU
外围设备
27
29
单片微机
纽扣电池
17
电源
电池

图 2

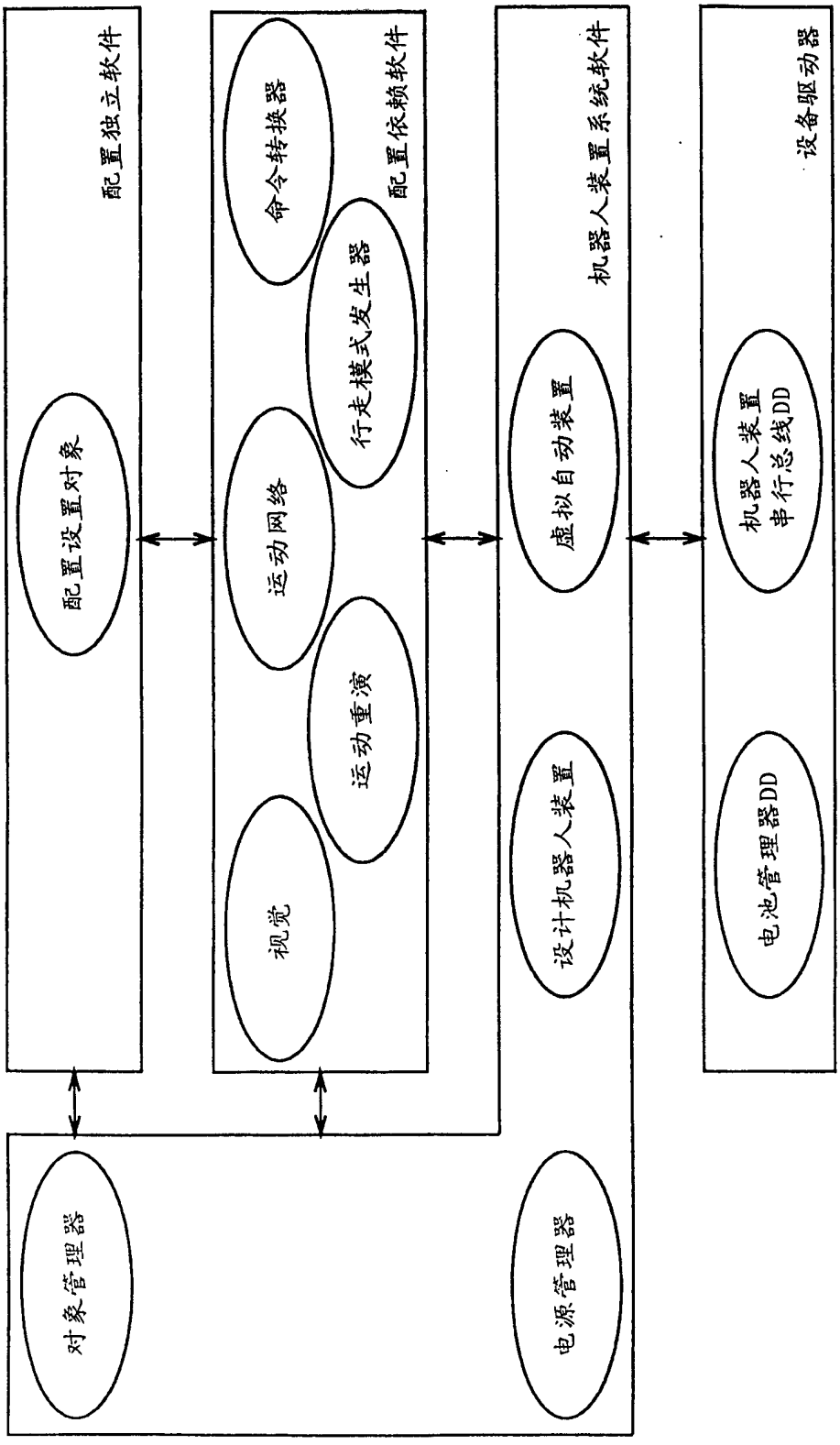

配置独立软件
配置设置对象
配置依赖软件
命令转换器
运动网络
行走模式发生器
运动重演
视觉
机器人装置系统软件
虚拟自动装置
设计机器人装置
设备驱动器
机器人装置
串行总线DD
电池管理器DD
对象管理器
电源管理器

图 3

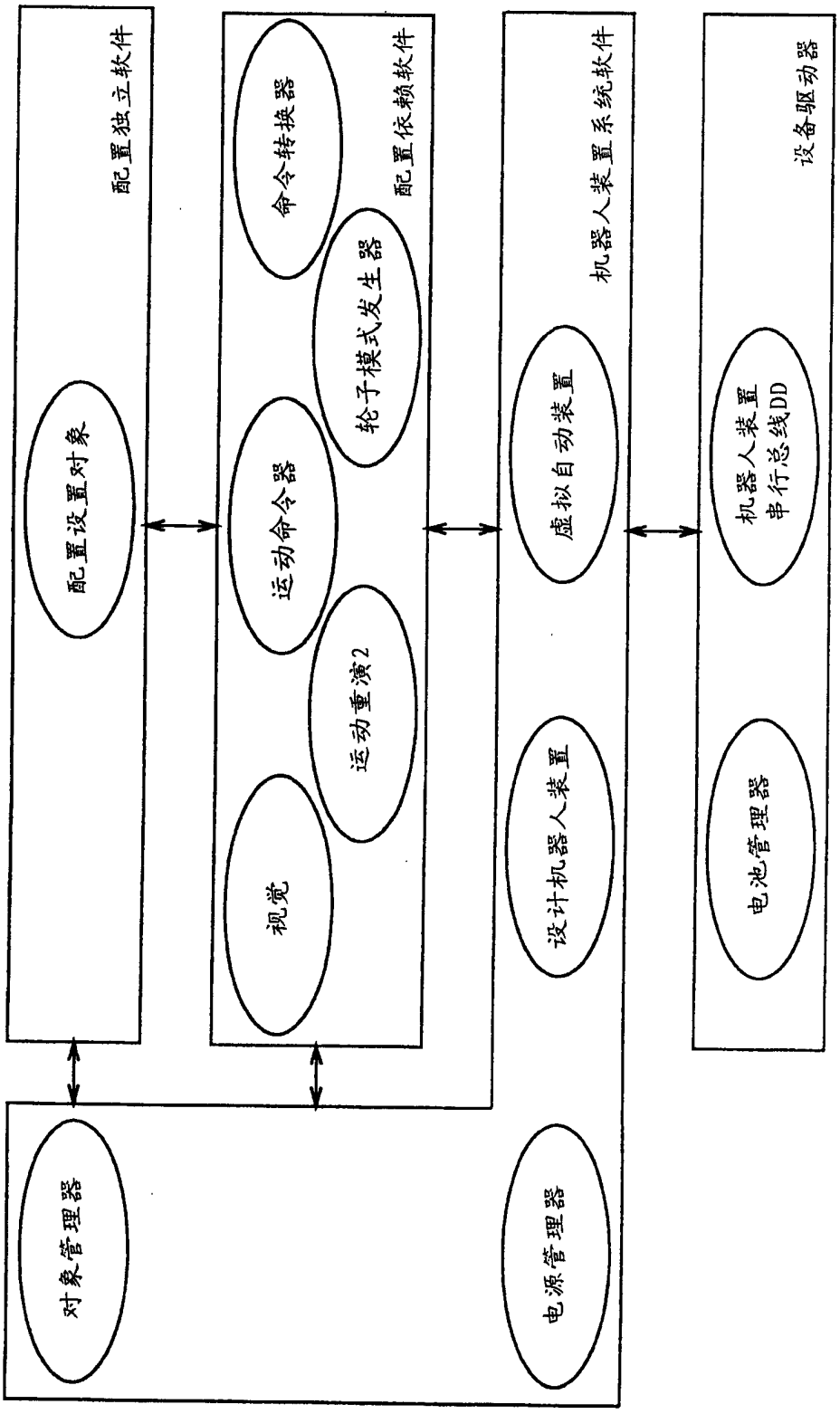

配置独立软件
配置设置对象
配置依赖软件
命令转换器
运动命令器
轮子模式发生器
运动重演2
视觉
机器人装置系统软件
虚拟自动装置
设计机器人装置
对象管理器
电源管理器
设备驱动器
机器人装置
串行总线DD
电池管理器

图 4

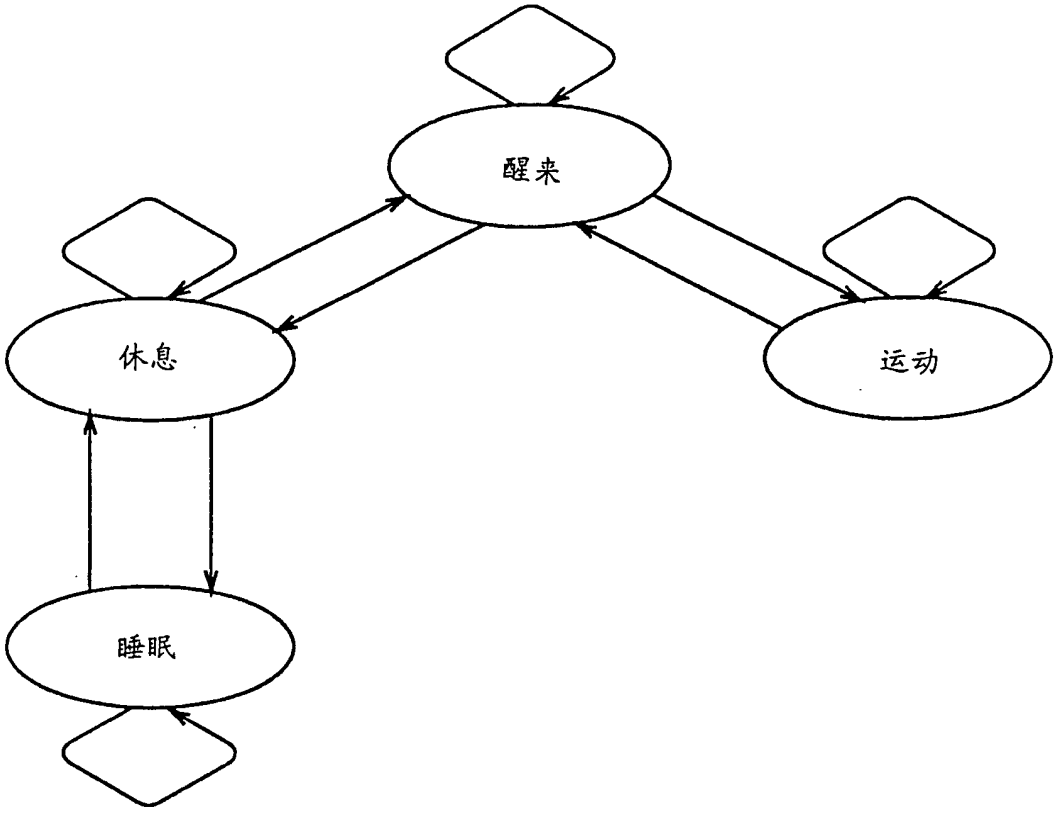
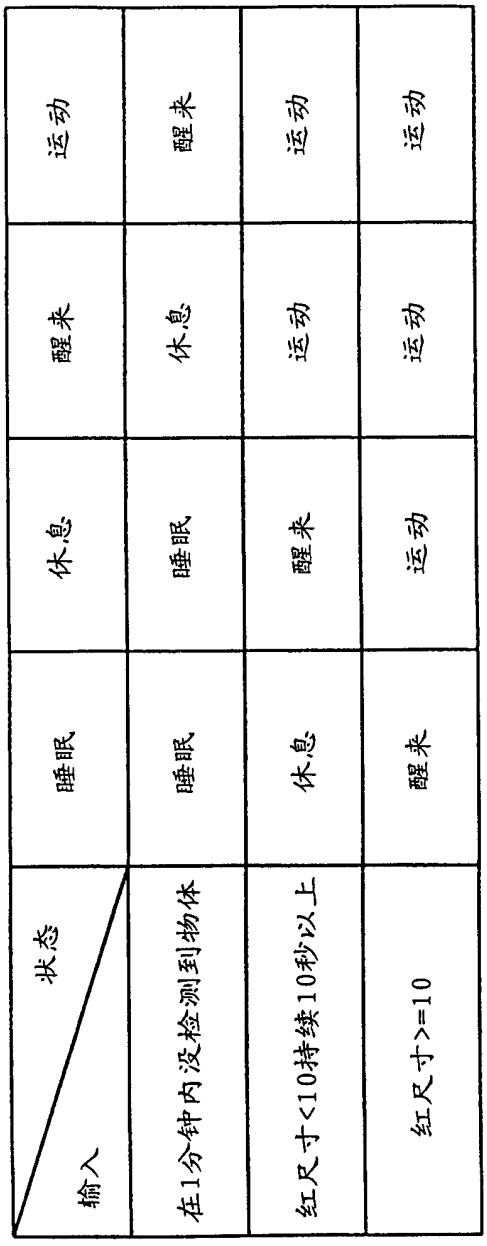


图 5



状态 / 输入	睡眠	休息	醒来	运动
在1分钟内没检测到物体	睡眠	睡眠	休息	醒来
红尺寸<10持续10秒以上	休息	醒来	运动	运动
红尺寸>=10	醒来	运动	运动	运动

图 6

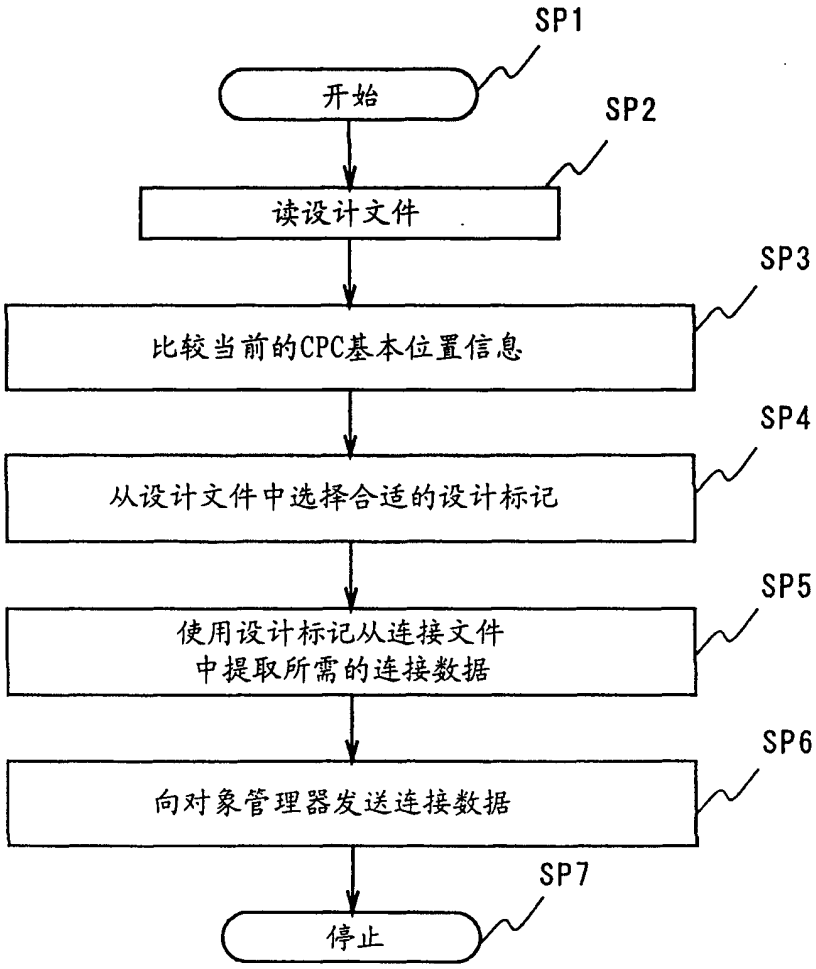

SP1
开始
SP2
读设计文件
SP3
比较当前的CPC基本位置信息
SP4
从设计文件中选择合适的设计标记
SP5
使用设计标记从连接文件
中提取所需的连接数据
SP6
向对象管理器发送连接数据
SP7
停止

图 7

```
DRX-4腿{
    DRX Head;
    DRX RightFrontLeg;
    DRX LeftFrontLeg;
    DRX RightRearLeg;
    DRX LeftRearLeg;
}
DRX轮{
    DRX Head;
    DRX RightFrontLeg;
    DRX LeftFrontLeg;
    DRX RightRearWheel;
    DRX LeftRearWheel;
}
DRX头{
    DRX CameraLink;
}
虚拟CameraLink{
    DRX CameraLink;
    Camera;
    Coordinate:/
}
DRX RightFrontLeg{
    Coordinate:/C1/C0-Joint;
    Coordinate:/C1/C0/C0-Joint;
    Coordinate:/C1/C0/C0/C0-Joint;
    Coordinate:/C1/C0/C0/C0/C0-TouchSensor;
}
DRX LeftFrontLeg{
    Coordinate:/C2/C0-Joint;
    Coordinate:/C2/C0/C0-Joint;
    Coordinate:/C2/C0/C0/C0-Joint;
    Coordinate:/C2/C0/C0/C0/C0-TouchSensor;
}
DRX RightRearLeg{
    Coordinate:/C3/C0-Joint;
    Coordinate:/C3/C0/C0-Joint;
    Coordinate:/C3/C0/C0/C0-Joint;
    Coordinate:/C3/C0/C0/C0/C0-TouchSensor;
}
DRX LeftRearLeg{
    Coordinate:/C4/C0-Joint;
    Coordinate;/C4/C0/C0-Joint;
    Coordinate:/C4/C0/C0/C0-Joint;
    Coordinate:/C4/C0/C0/C0/C0-TouchSensor;
}
DRX RightRearWheel{
    Coordinate:/C3/C0-wheel;
}
DRX LeftRearWheel{
    Coordinate:/C4/C0-wheel;
}
```

图 8

```
CPC 当前信息{
    Coordinate:/C0/C0-Joint;
    Coordinate:/C0/C0/C0-Joint;
    Coordinate:/C0/C0/C0/C0-Joint;
    Coordinate:/C0/C0/C/C0/C0-Camera;
    Coordinate:/C1/C0-Joint;
    Coordinate:/C1/C0/C0-Joint;
    Coordinate:/C1/C0/C0/C0-Joint;
    Coordinate:/C1/C0/C0/C0/C0-TouchSensor;
    Coordinate:/C2/C0-Joint;
    Coordinate:/C2/C0/C0-Joint;
    Coordinate:/C2/C0/C0/C0-Joint;
    Coordinate:/C2/C0/C0/C0/C0-TouchSensor;
    Coordinate:/C3/C0-Joint;
    Coordinate:/C3/C0/C0-Joint;
    Coordinate:/C3/C0/C0/C0-Joint;
    Coordinate:/C3/C0/C0/C0/C0-TouchSensor;
    Coordinate:/C4/C0-Joint;
    Coordinate:/C4/C0/C0-Joint;
    Coordinate:/C4/C0/C0/C0-Joint;
    Coordinate:/C4/C0/C0/C0/C0-TouchSensor;
```

图 9

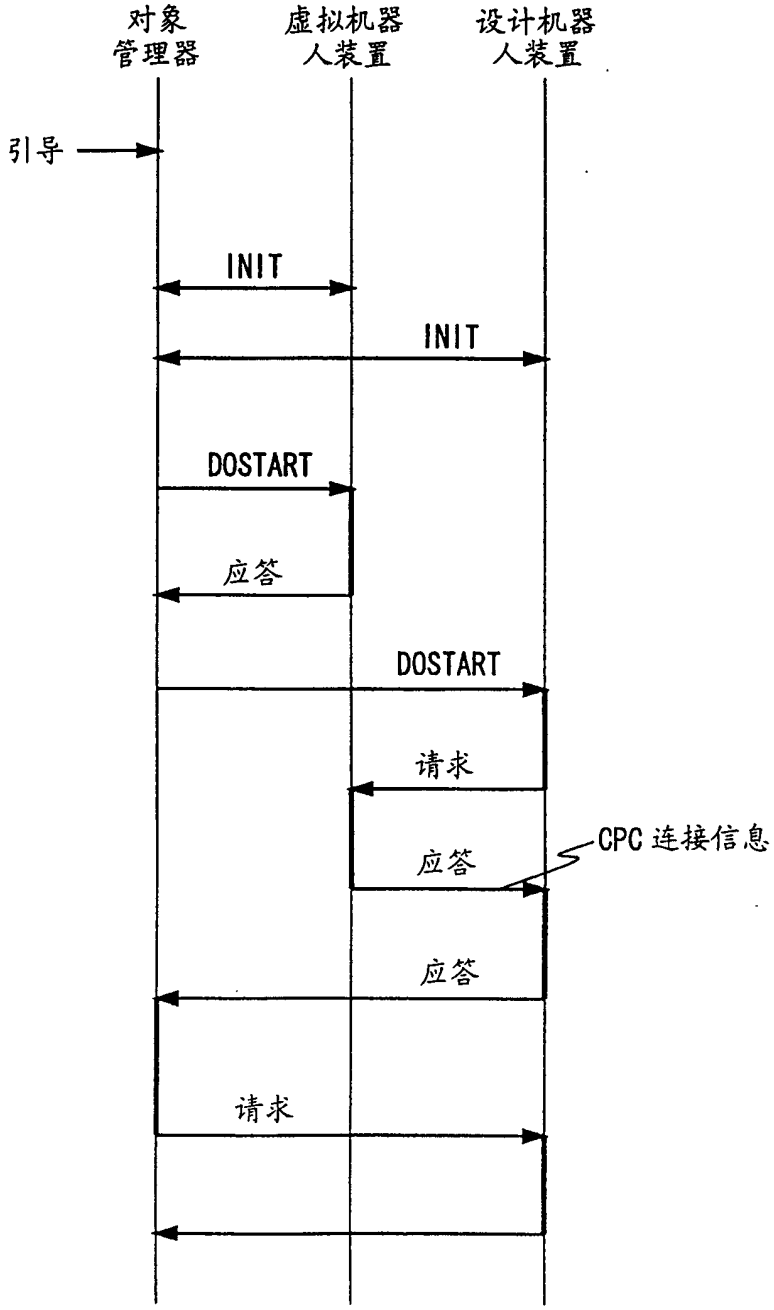

对象
管理器
虚拟机器
人装置
设计机器
人装置
引导
INIT
INIT
DOSTART
应答
DOSTART
请求
CPC 连接信息
应答
应答
请求

图 10

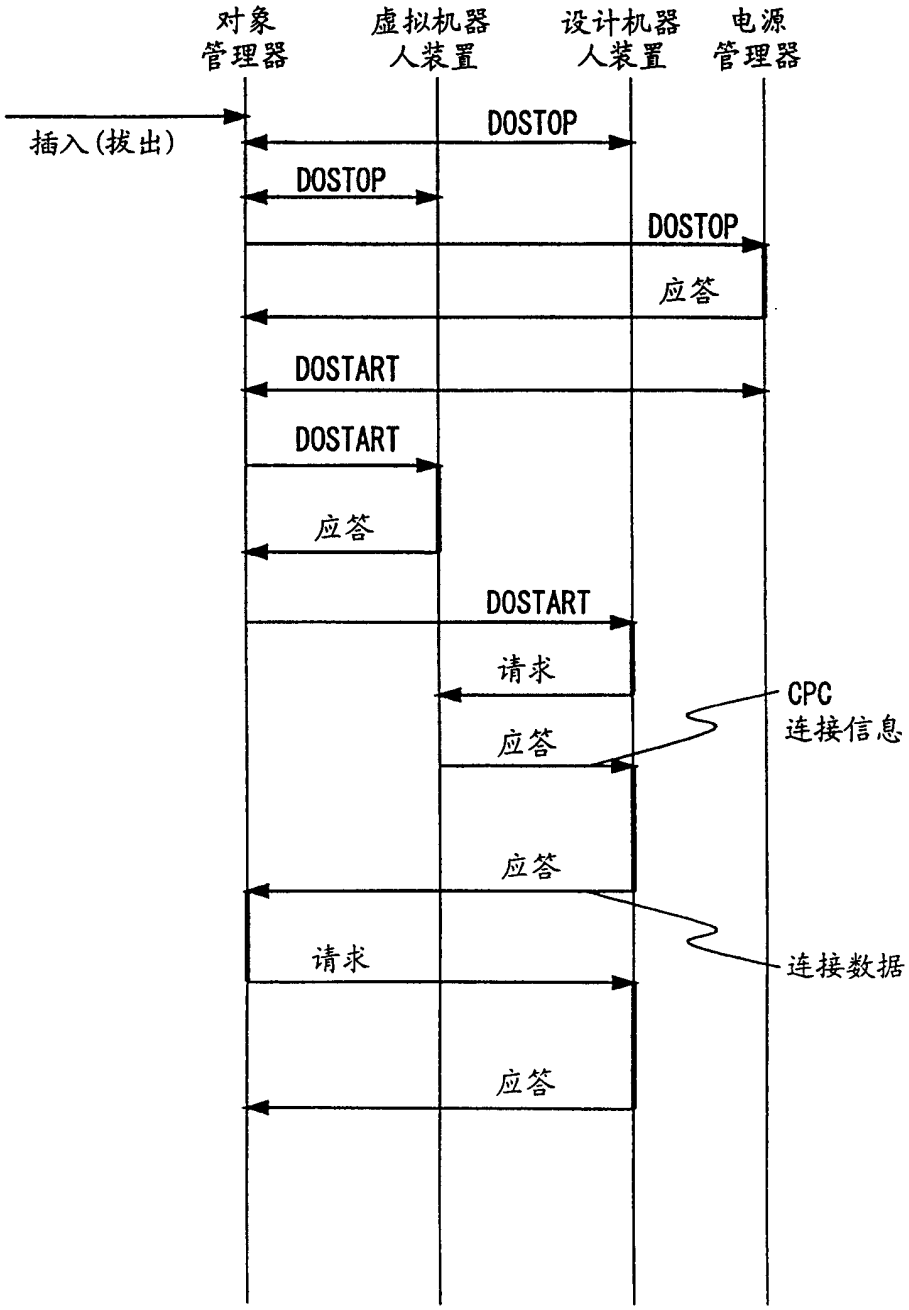

对象
管理器
虚拟机器
人装置
设计机器
人装置
电源
管理器
插入(拔出)
DOSTOP
DOSTOP
DOSTOP
应答
DOSTART
DOSTART
应答
DOSTART
请求
CPC
连接信息
应答
应答
请求
连接数据
应答

图 11

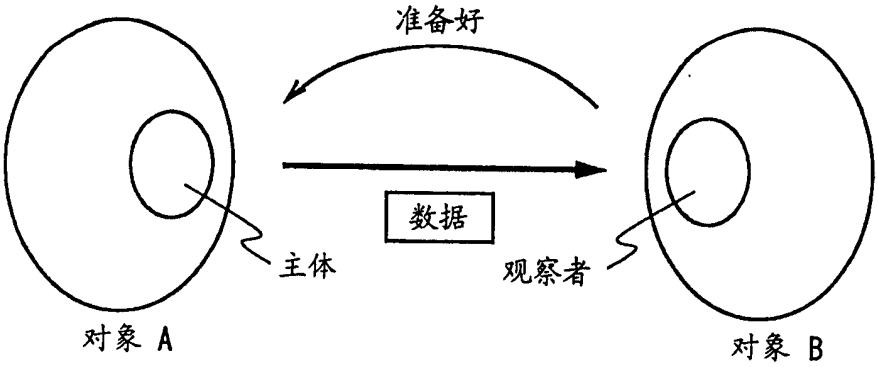


图 12

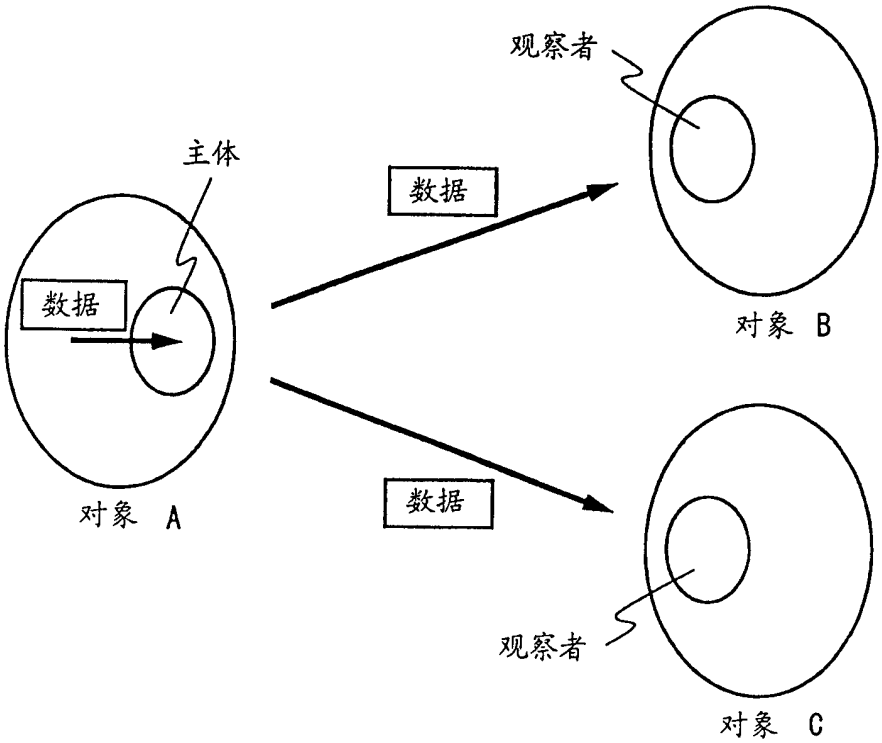


图 13

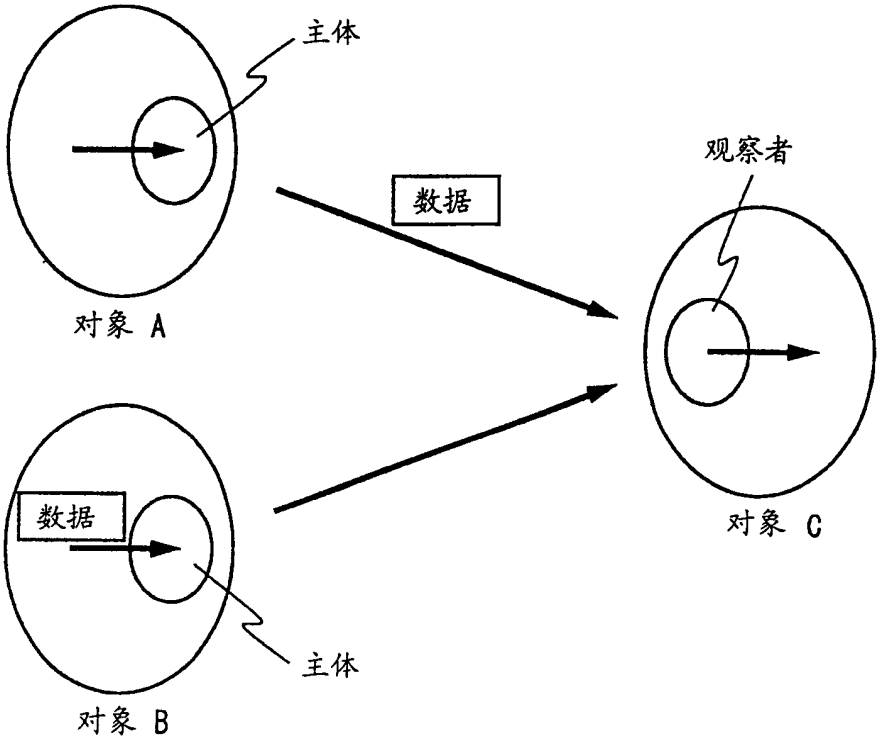


图 14

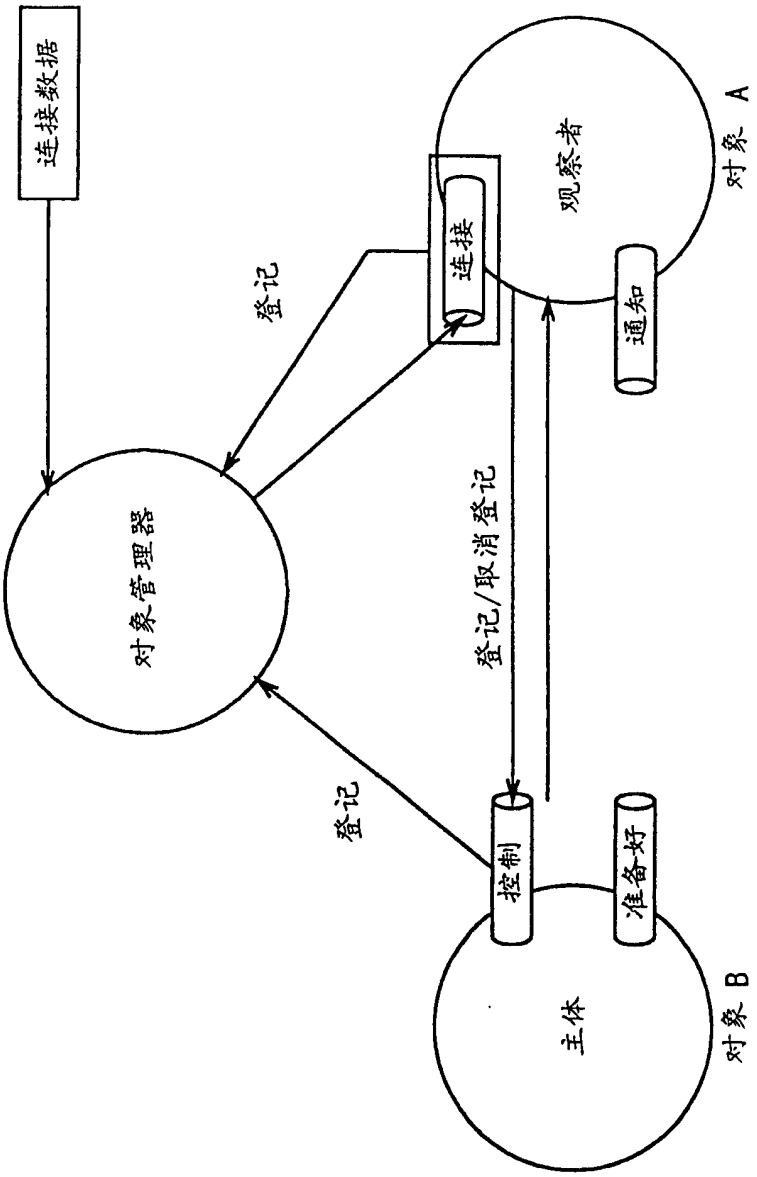


图 15

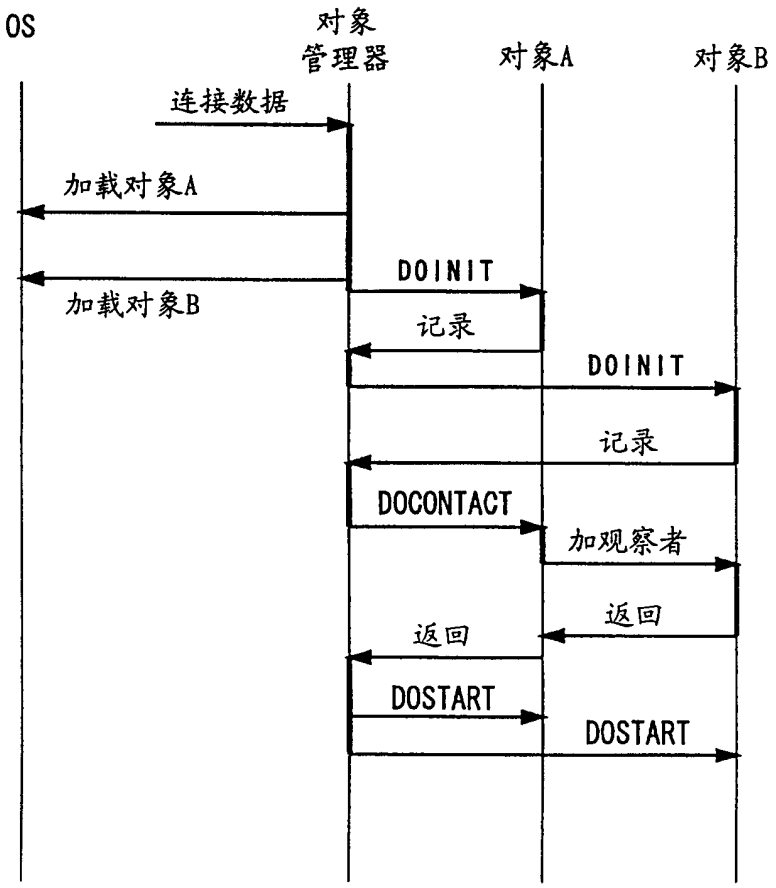

OS
对象
管理器
对象A
对象B
连接数据
加载对象A
DOINIT
加载对象B
记录
DOINIT
记录
DOCONTACT
加观察者
返回
返回
DOSTART
DOSTART

图 16

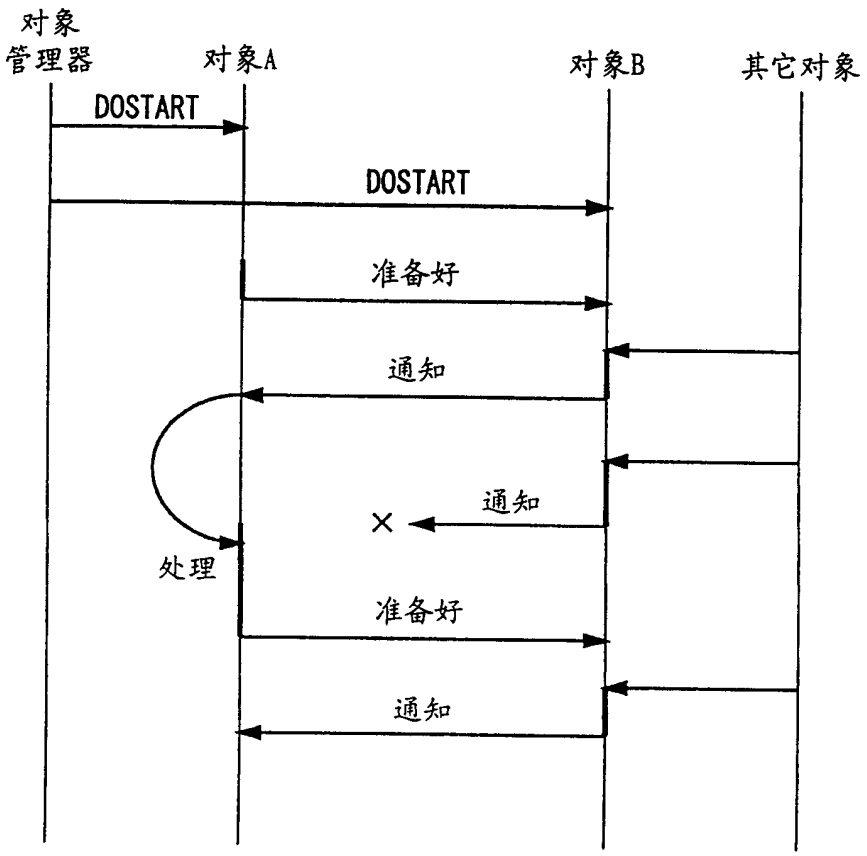


图 17

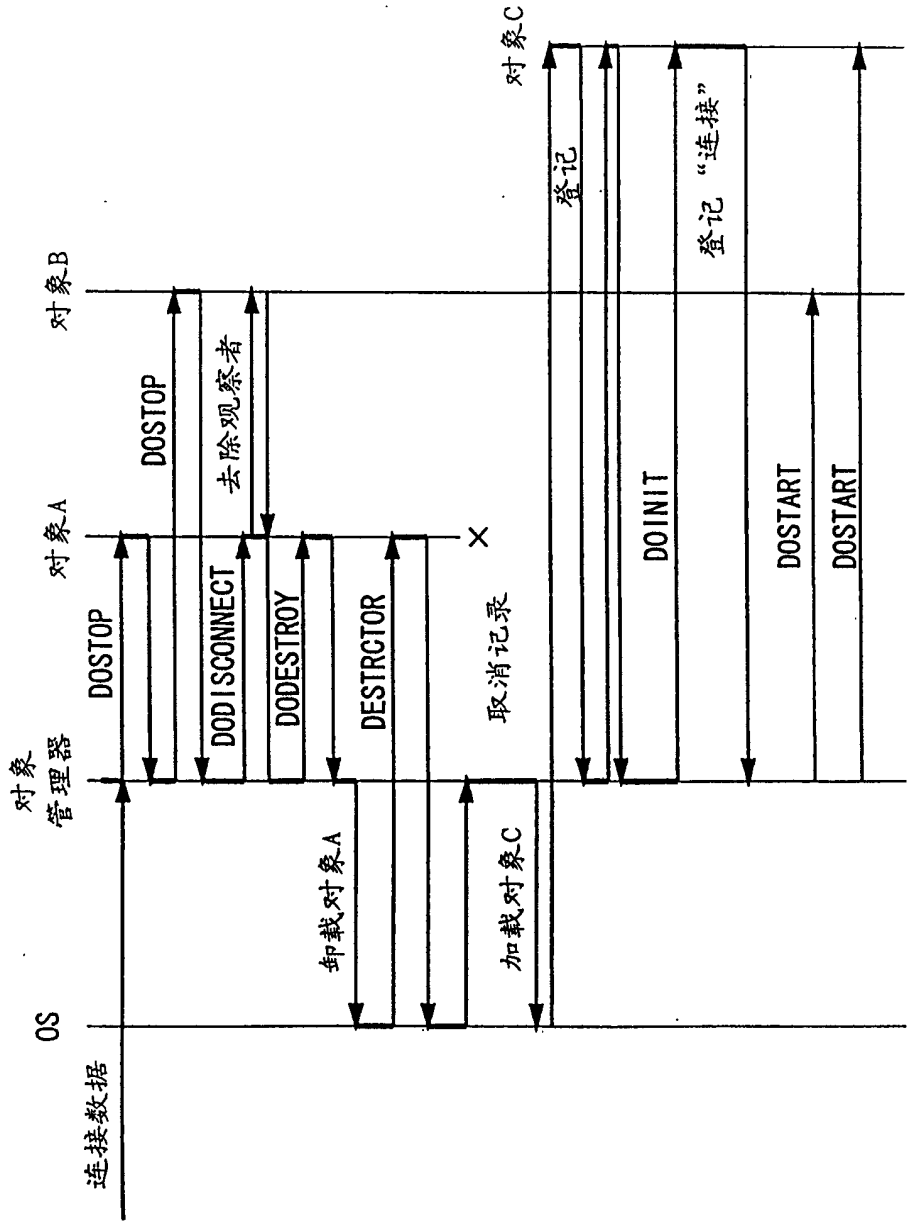

连接数据
OS
对象
管理器
对象A
对象B
对象C
DOSTOP
DOSTOP
DODISCONNECT
去除观察者
DODESTROY
DESTRCTOR
卸载对象A
取消记录
加载对象C
登记
DOINIT
登记“连接”
DOSTART
DOSTART

图 18

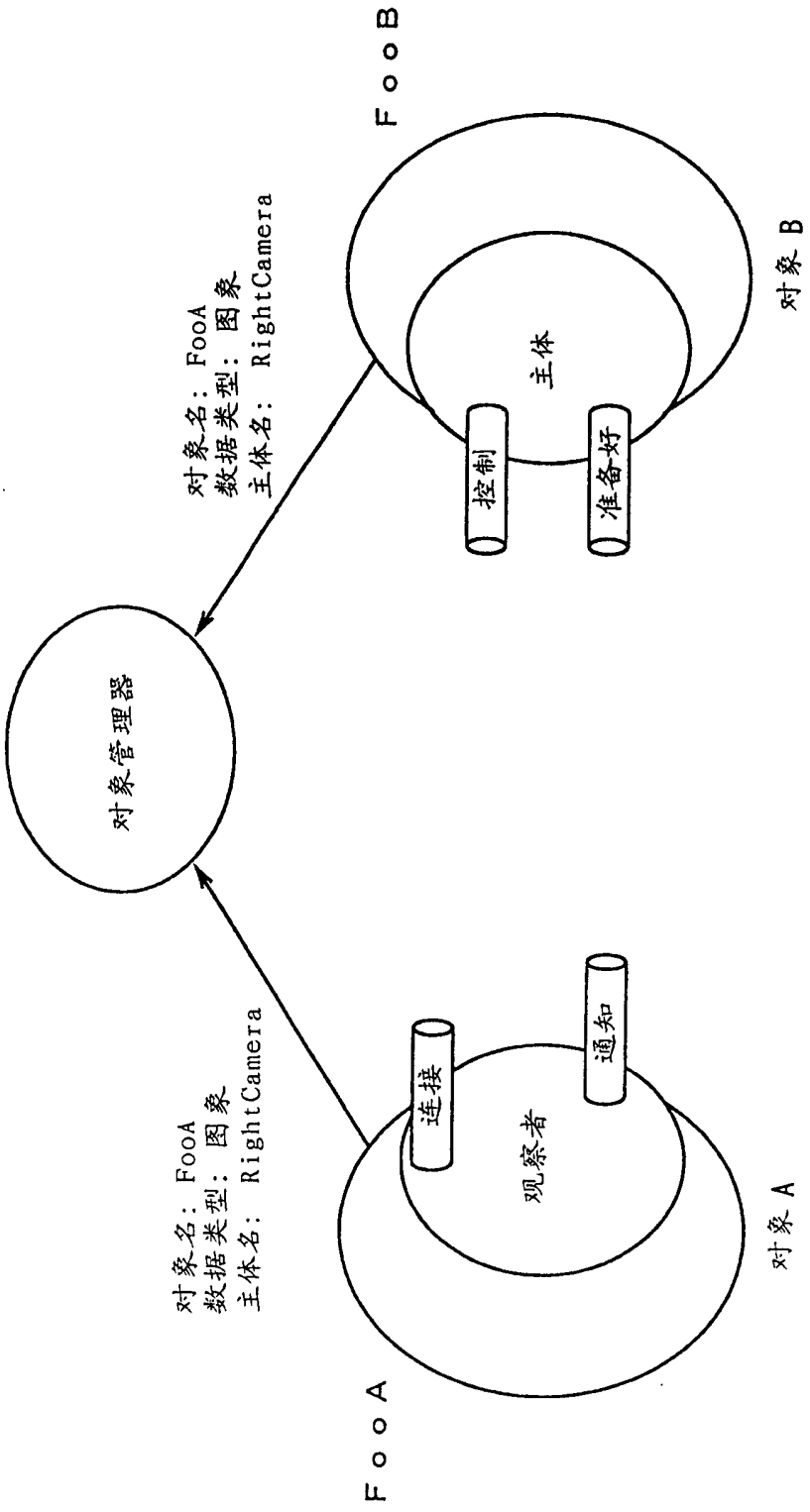

FooB
对象B
主体
控制
准备好
对象管理器
对象名：FooA
数据类型：图象
主体名：RightCamera
对象名：FooA
数据类型：图象
主体名：RightCamera
FooA
对象A
观察者
连接
通知

图 19

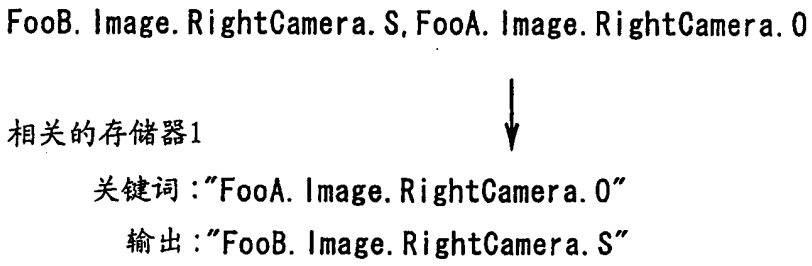

FooB. Image. RightCamera. S, FooA. Image. RightCamera. 0
相关的存储器1
关键词："FooA. Image. RightCamera. 0"
输出："FooB. Image. RightCamera. S"

图 20


FooB. Image. RightCamera、1, 5. S
相关的存储器2
关键词：FooB. Image. RightCamera. S
输出：1, 5
FooA. Image. RightCamera、2, 5. 0
相关的存储器2
关键词：FooA. Image. RightCamera. 0
输出：2, 5

图 21

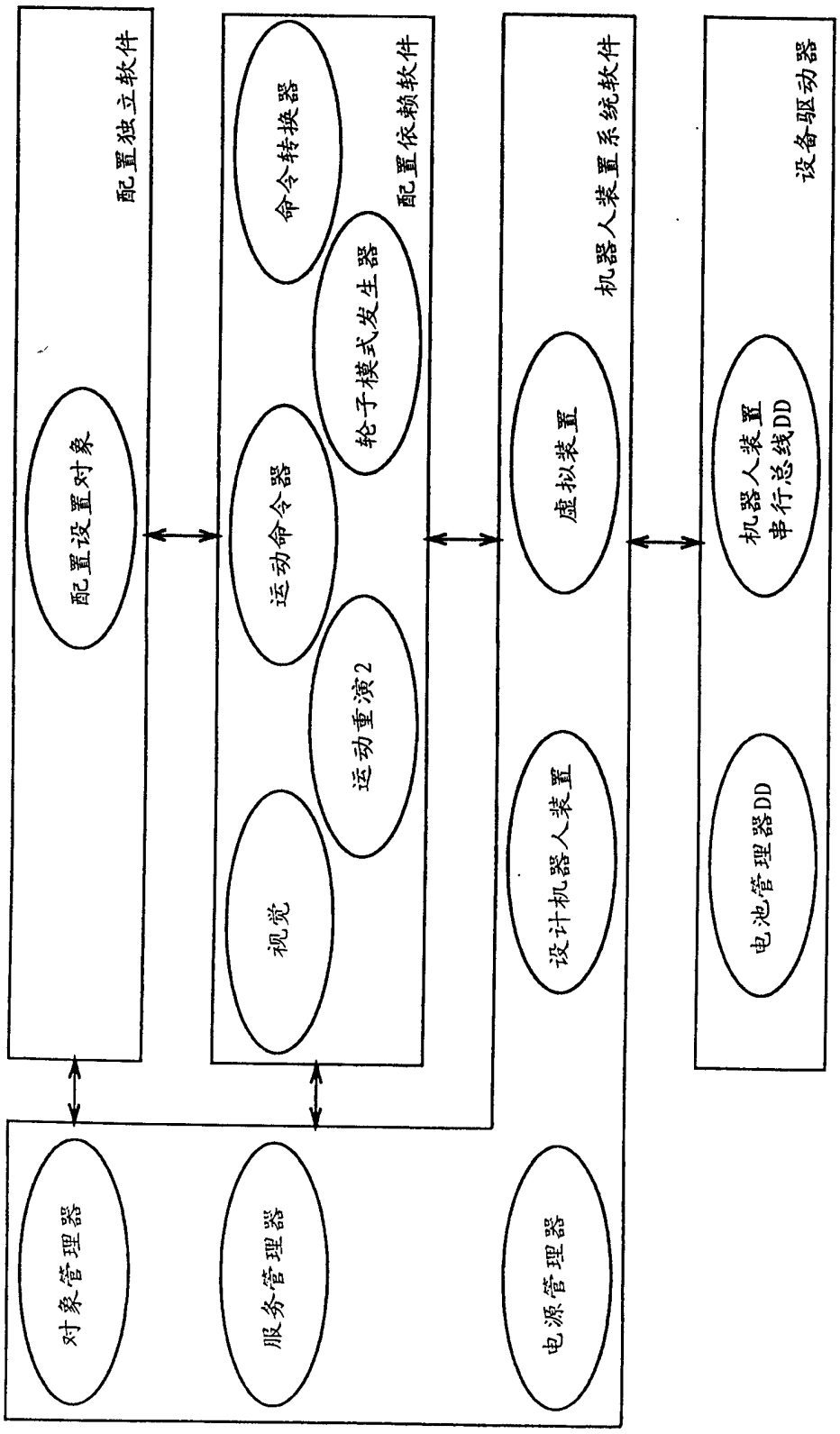

配置独立软件
配置设置对象
配置依赖软件
命令转换器
运动命令器
轮子模式发生器
运动重演2
视觉
机器人装置系统软件
虚拟装置
设计机器人装置
设备驱动器
机器人装置
串行总线DD
电池管理器DD
对象管理器
服务管理器
电源管理器

图 22

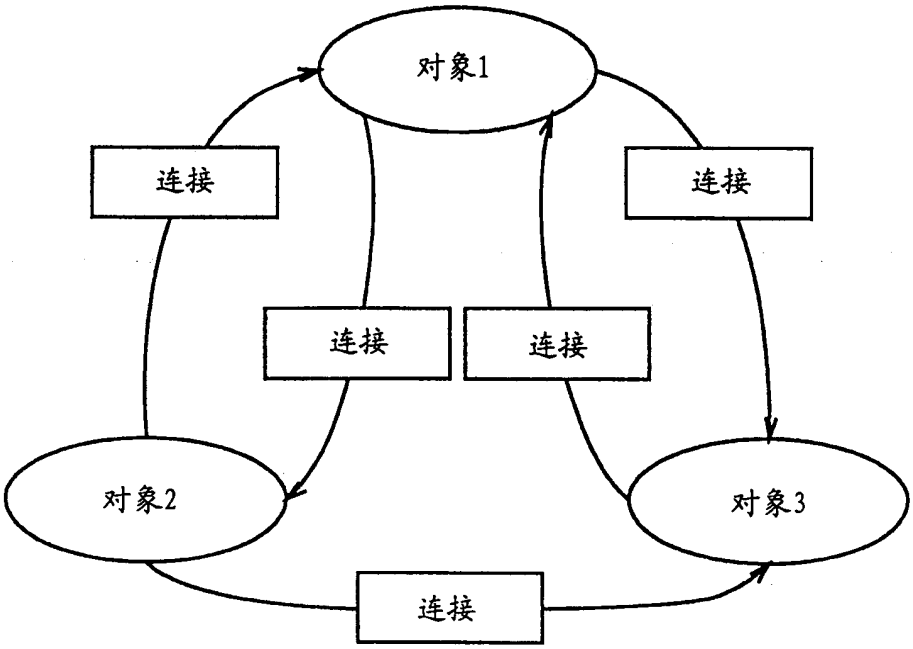


图 23

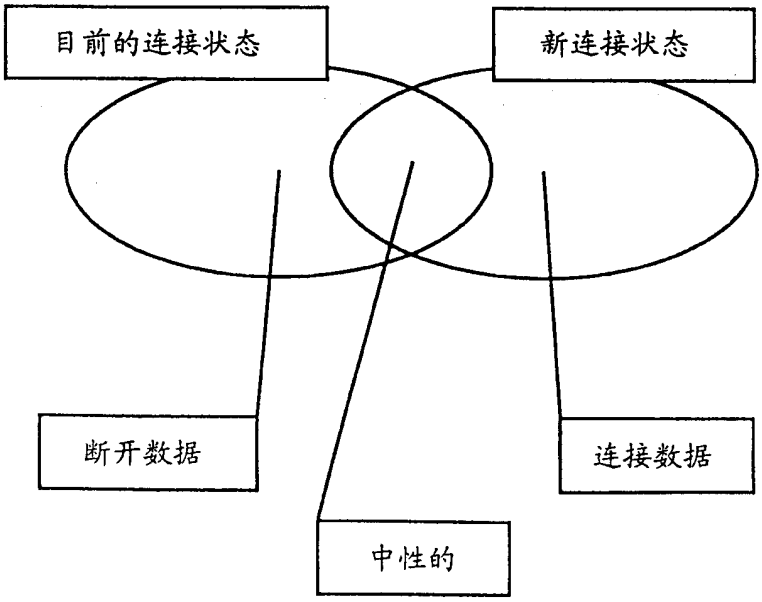


图 24

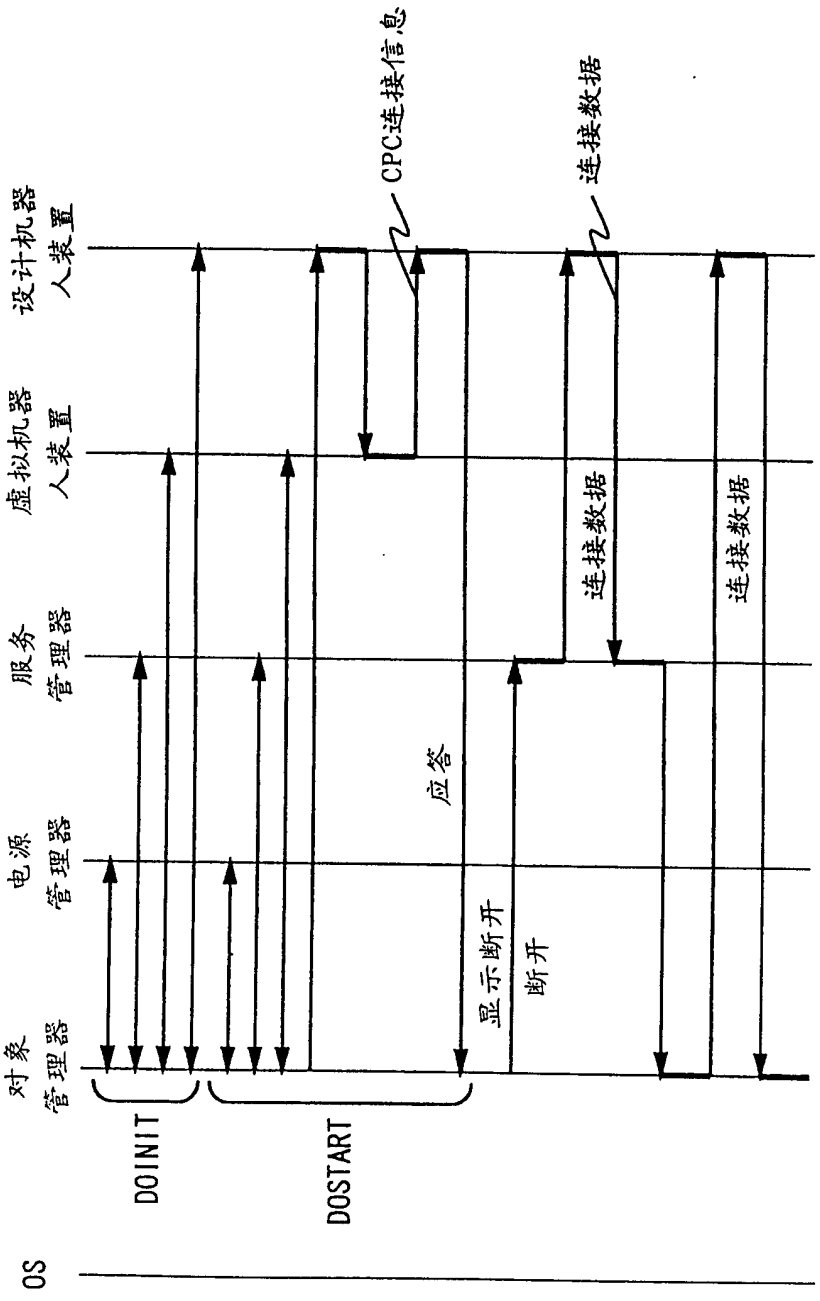

OS
对象管理器
电源管理器
服务管理器
虚拟机器人装置
设计机器人装置
DOINIT
DOSTART
应答
显示断开
断开
CPC连接信息
连接数据
连接数据
连接数据

图 25

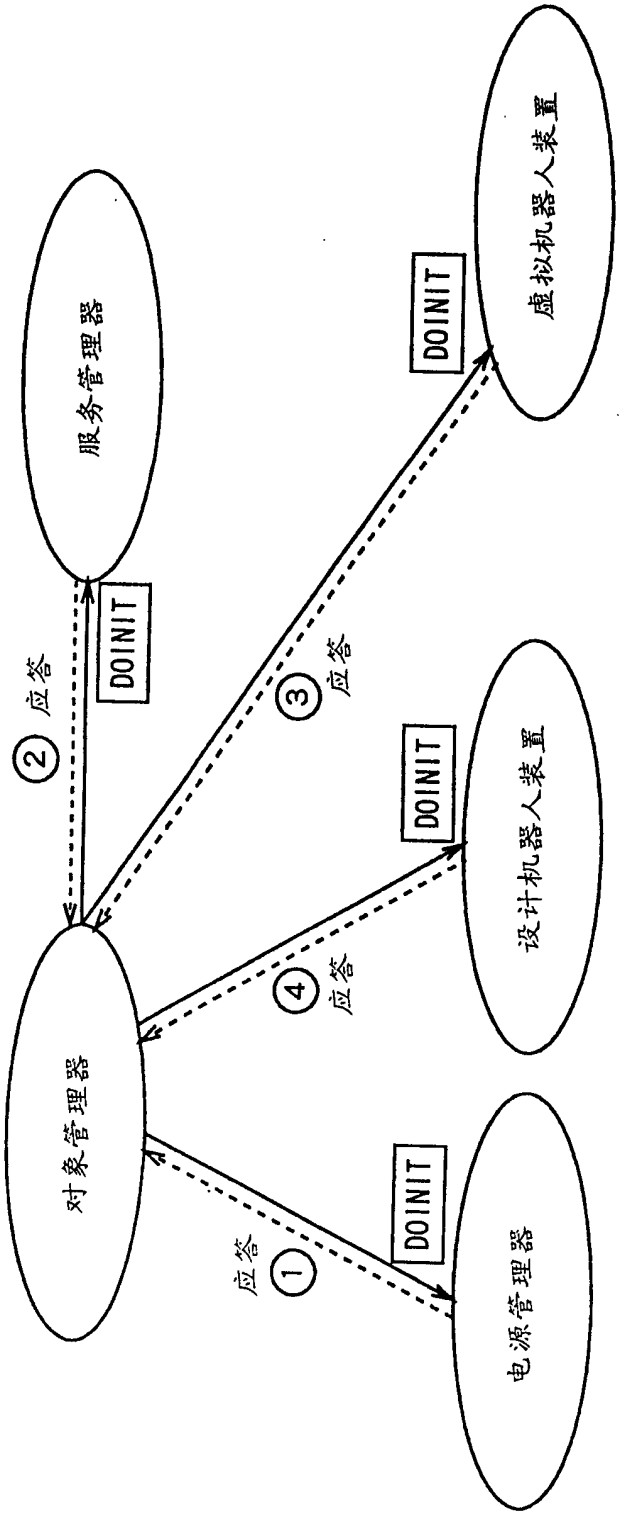

服务管理器
虚拟机器人装置
DOINIT
② 应答
DOINIT
③ 应答
对象管理器
设计机器人装置
DOINIT
④ 应答
应答 ①
DOINIT
电源管理器

图 26

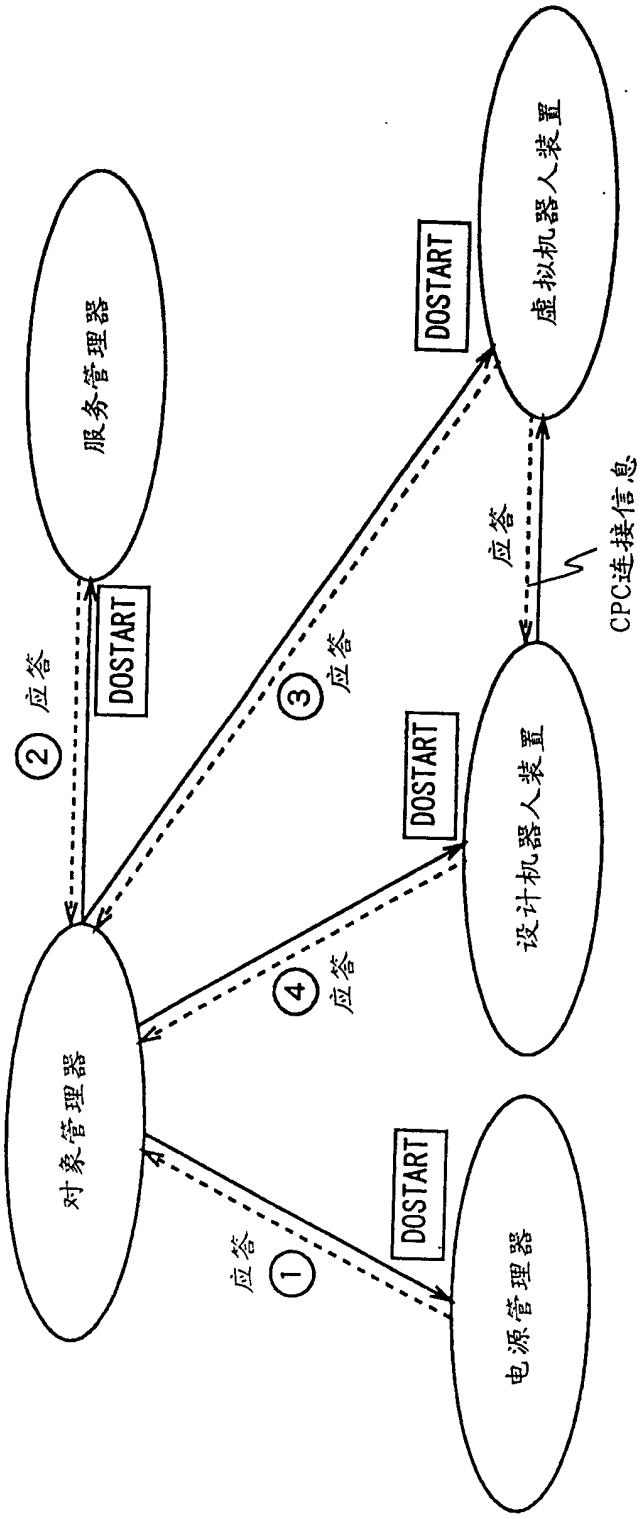

服务管理器
虚拟机器人装置
对象管理器
设计机器人装置
电源管理器
DOSTART
DOSTART
DOSTART
DOSTART
① 应答
② 应答
③ 应答
④ 应答
应答
CPC连接信息

图 27

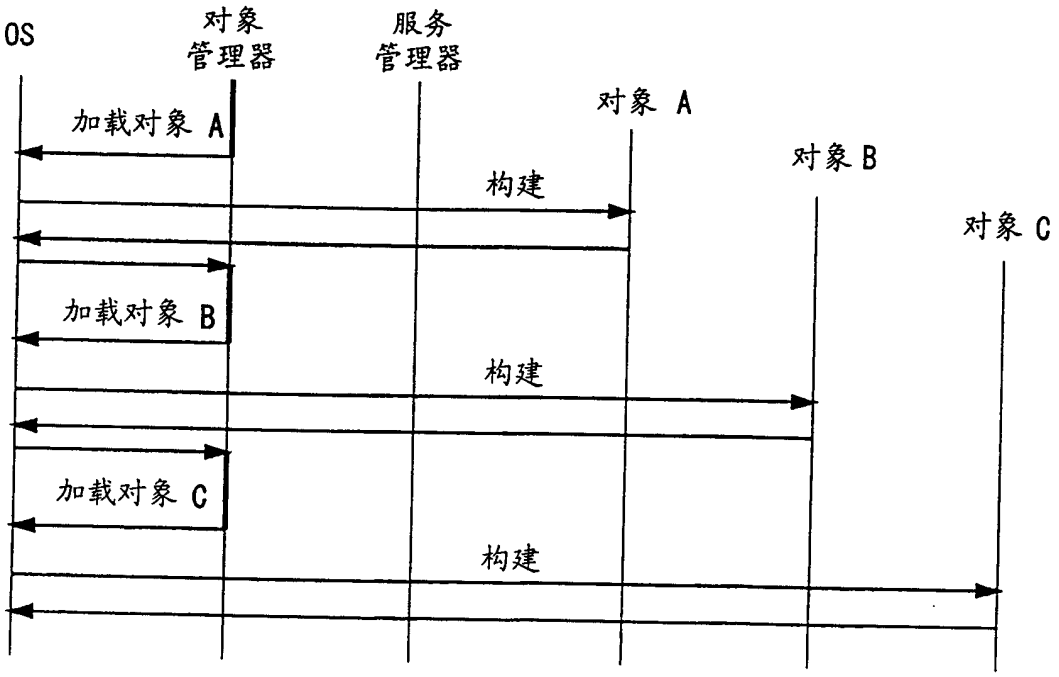


图 28

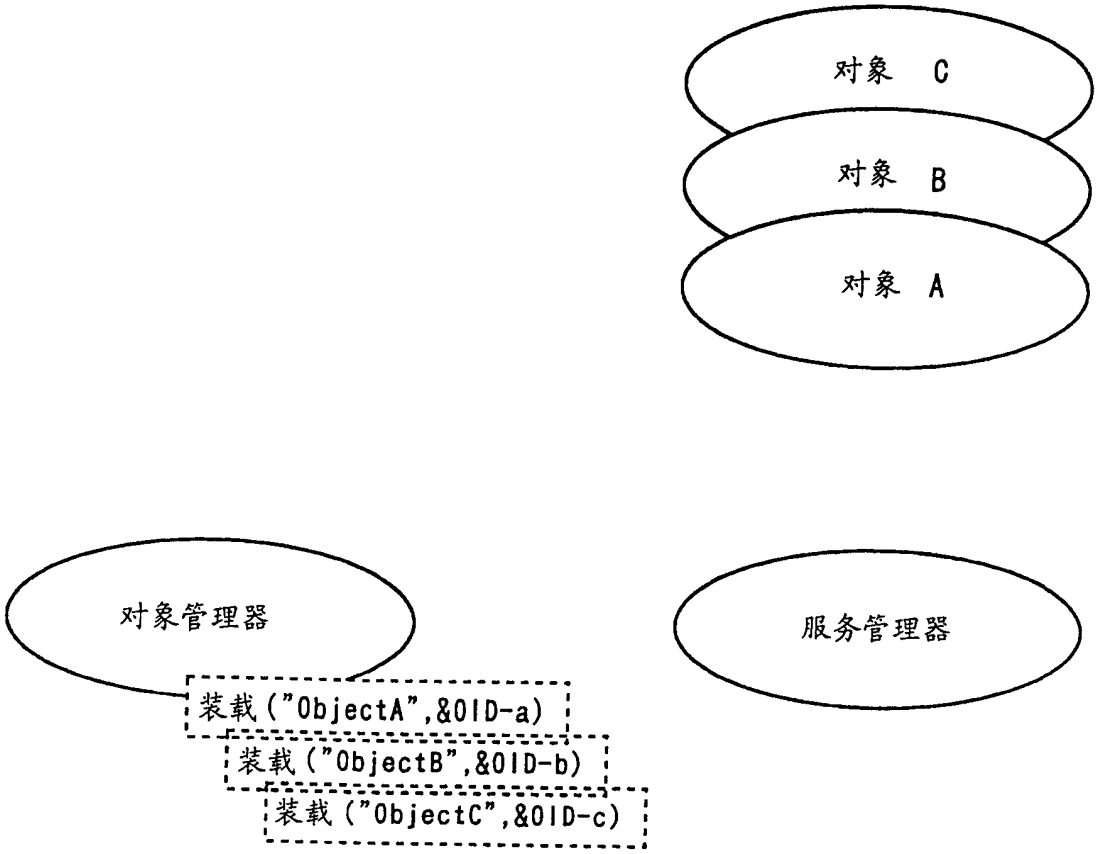

对象 C
对象 B
对象 A
对象管理器
服务管理器
装载("ObjectA",&OID-a)
装载("ObjectB",&OID-b)
装载("ObjectC",&OID-c)

图 29

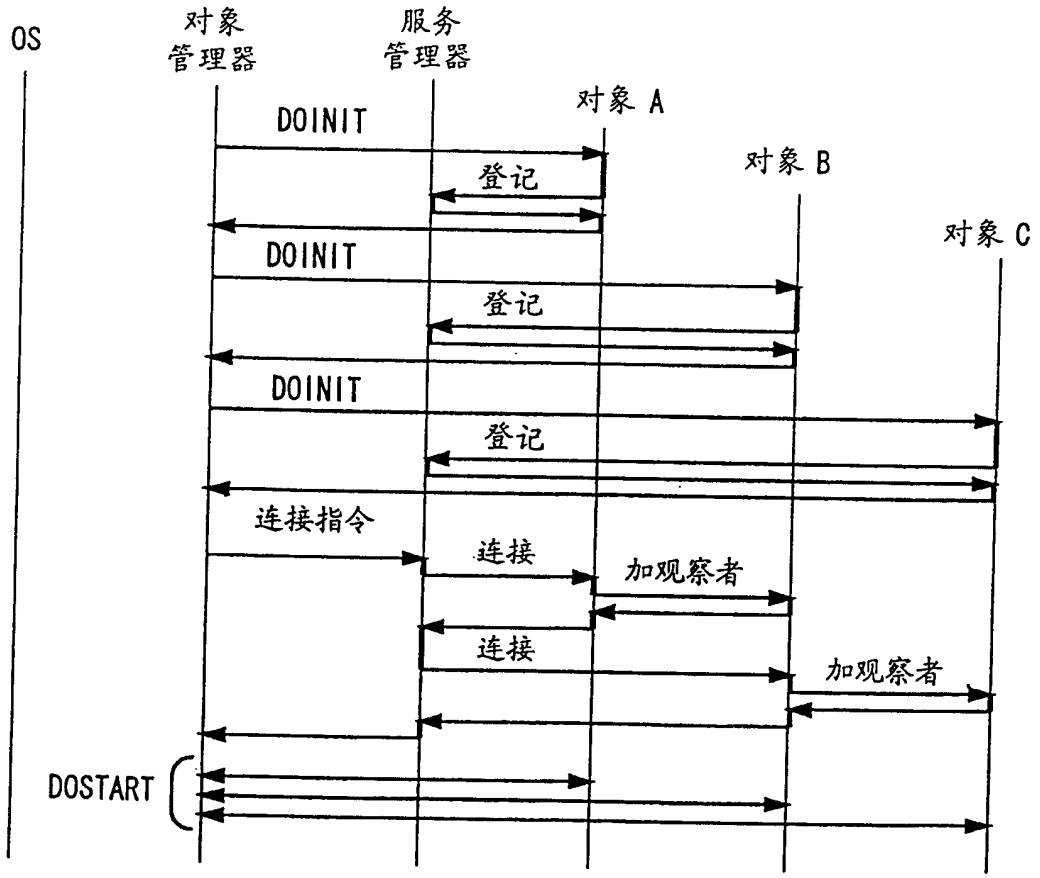

OS
对象
管理器
服务
管理器
对象 A
对象 B
对象 C
DOINIT
登记
DOINIT
登记
DOINIT
登记
连接指令
连接
加观察者
连接
加观察者
DOSTART

图 30

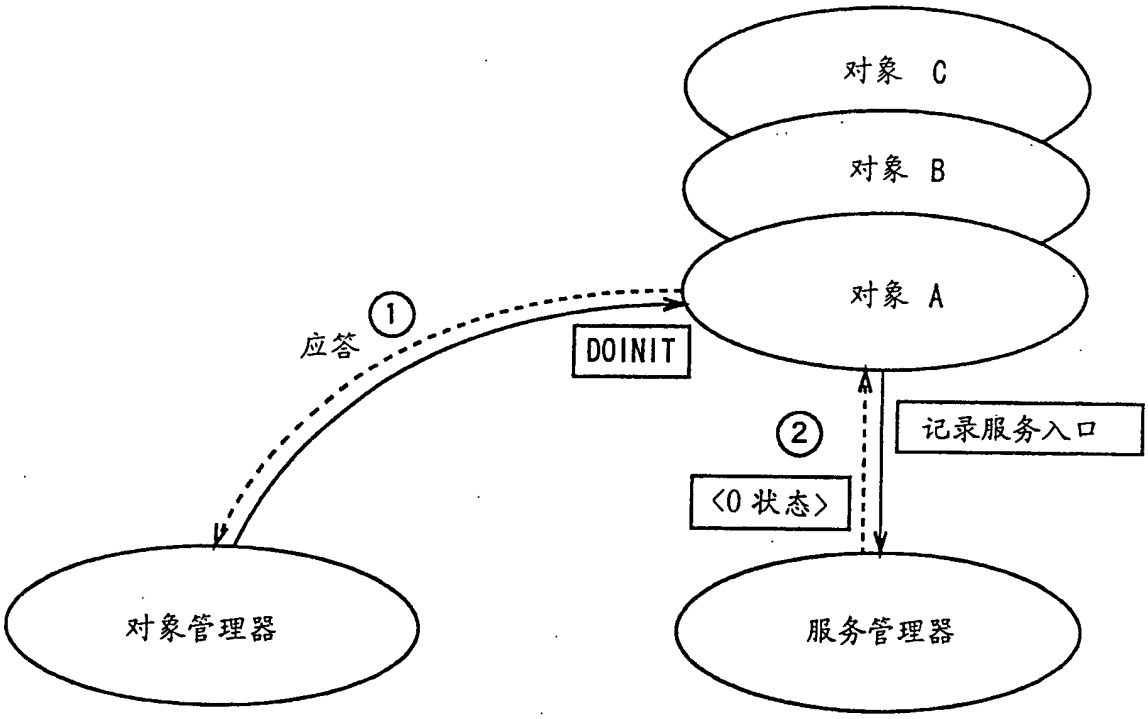


图 31

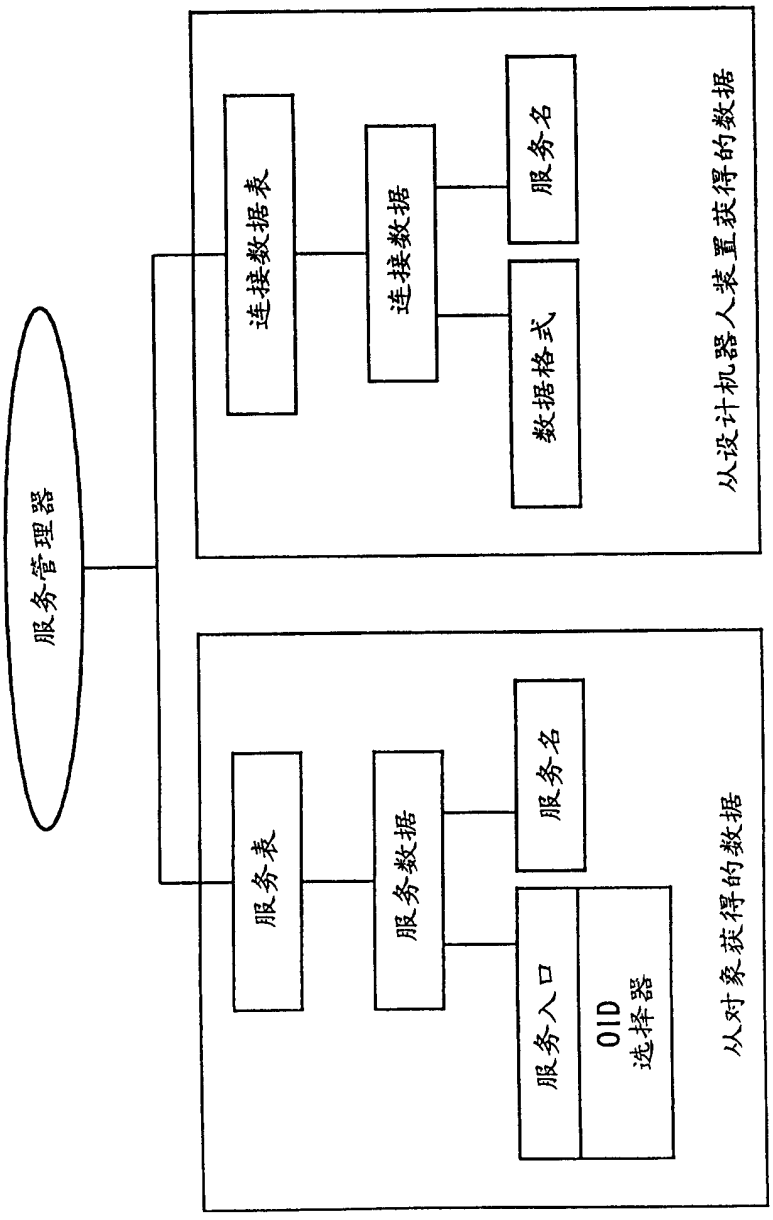

服务管理器
从设计机器人装置获得的数据
连接数据表
连接数据
服务名
数据格式
从对象获得的数据
服务表
服务数据
服务名
服务入口
OID
选择器

图 32

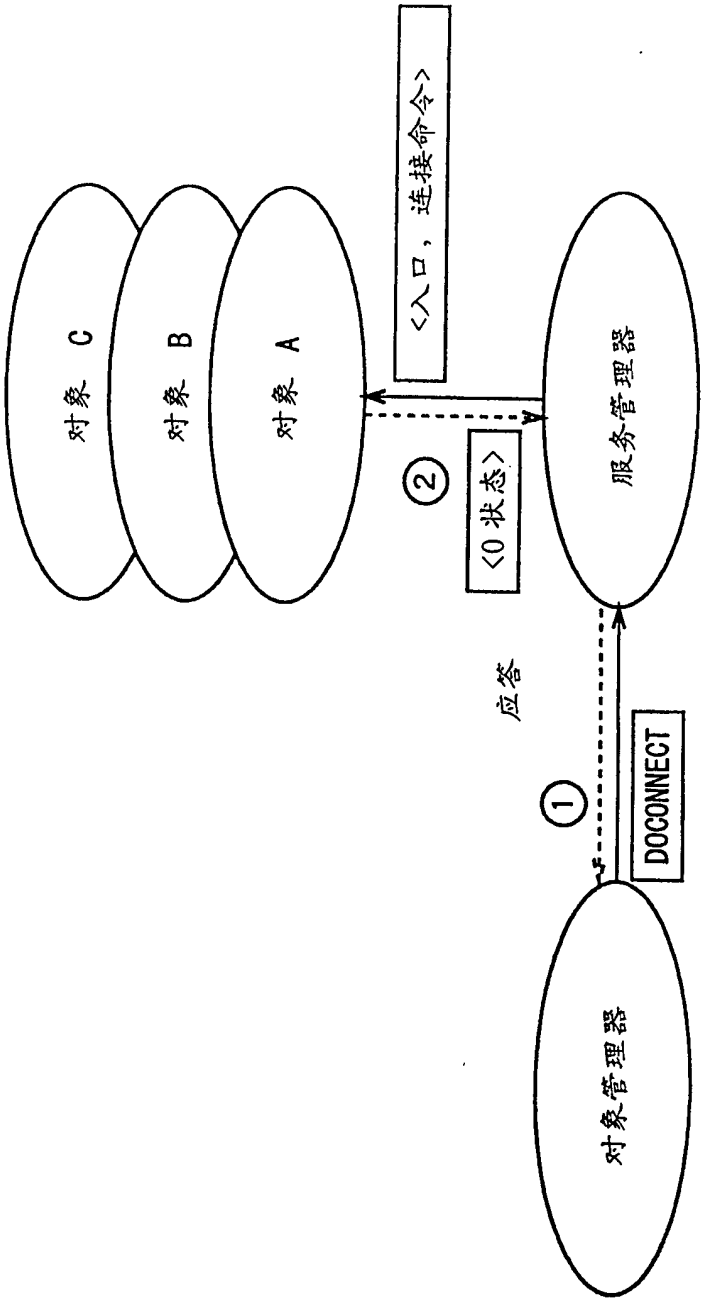


图 33

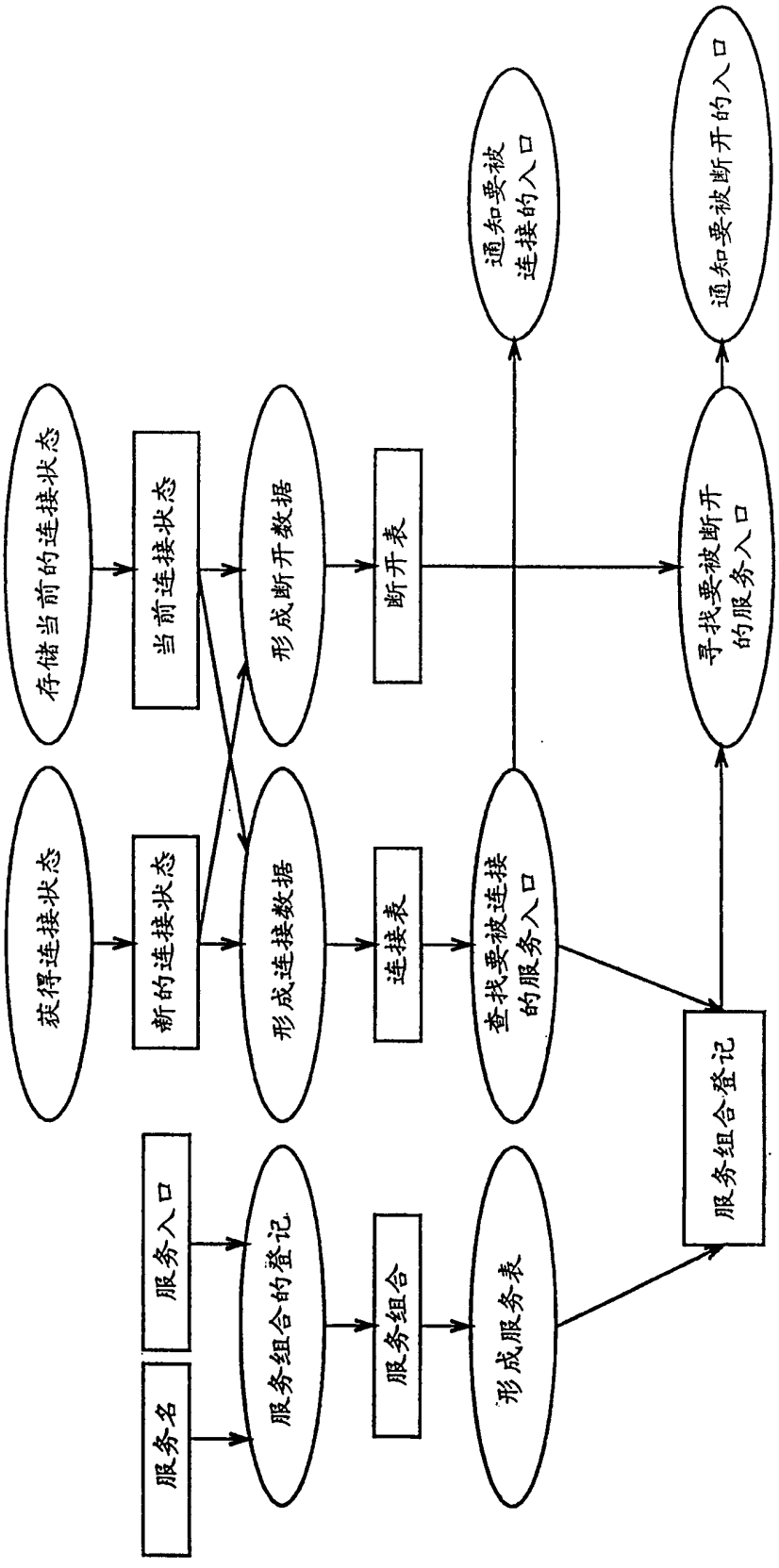

服务名
服务入口
服务组合的登记
服务组合
形成服务表
服务组合登记
获得连接状态
断的连接状态
形成连接数据
连接表
查找要被连接的服务入口
存储当前的连接状态
当前连接状态
形成断开数据
断开表
通知要被连接的入口
寻找要被断开的服务入口
通知要被断开的入口

图 34

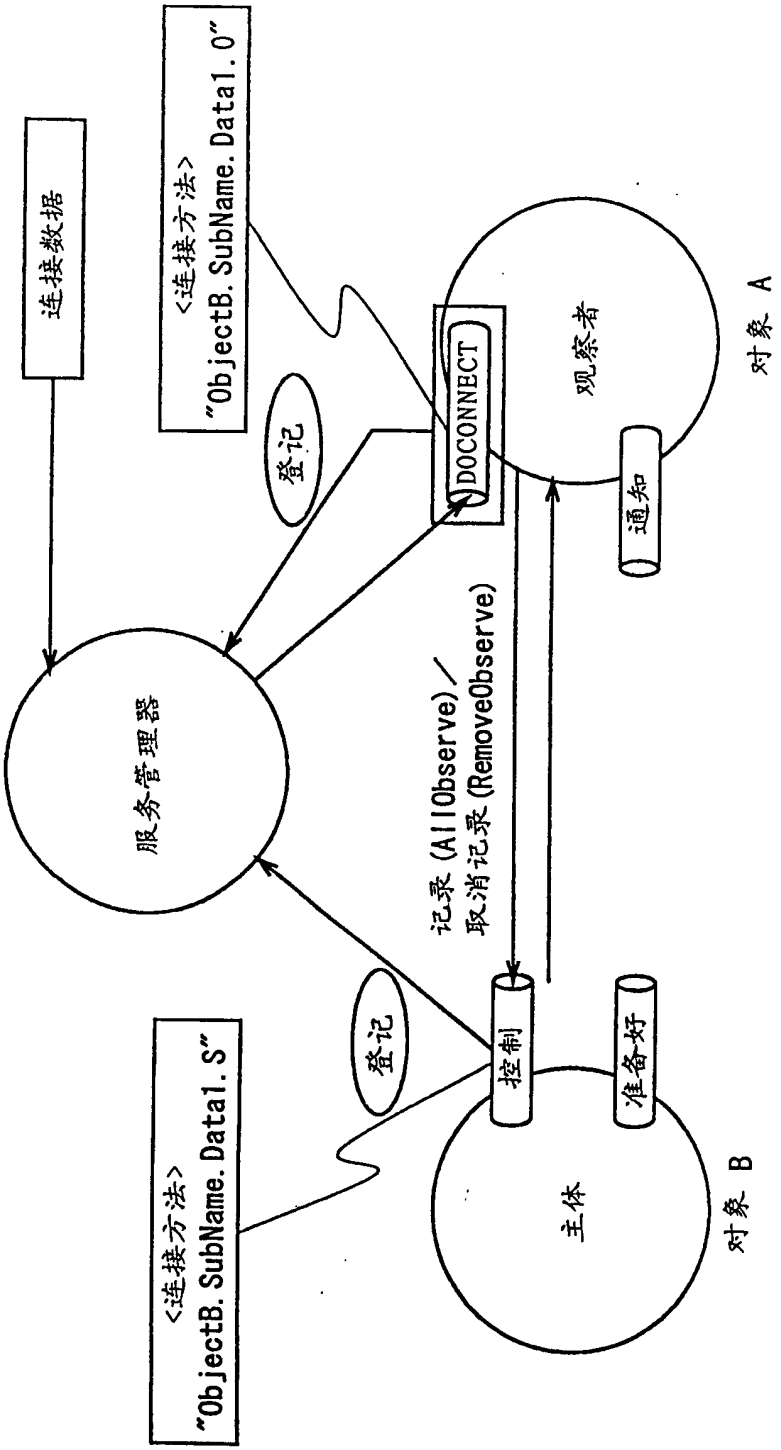

连接数据
<连接方法>
"ObjectB.SubName.Data1.0"
登记
DOCONNECT
观察者
对象 A
通知
服务管理器
记录(AllObserve)／
取消记录(RemoveObserve)
登记
<连接方法>
"ObjectB.SubName.Data1.S"
控制
准备好
主体
对象 B

图 35

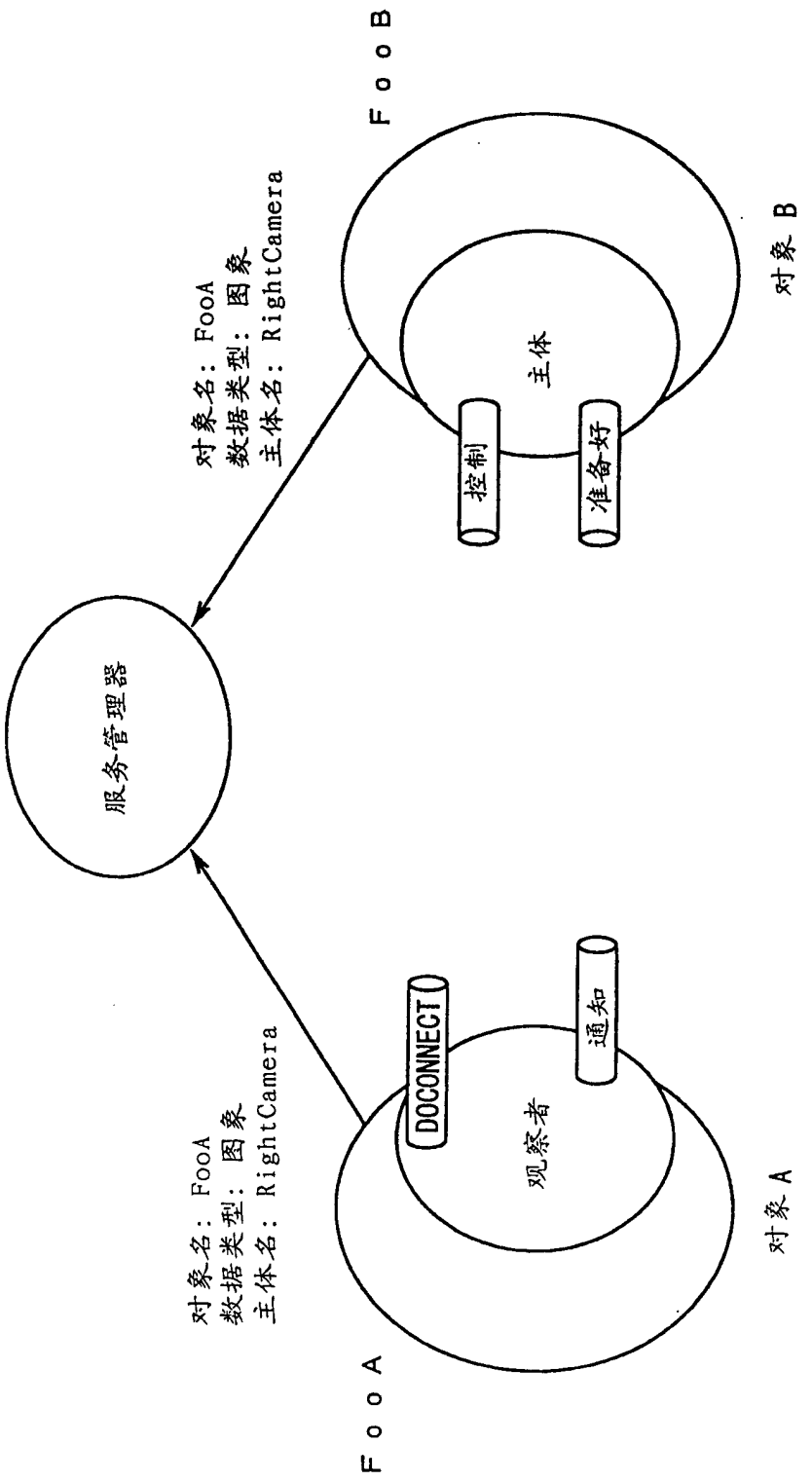

服务管理器
对象名：FooA
数据类型：图象
主体名：RightCamera
对象名：FooA
数据类型：图象
主体名：RightCamera
FooB
对象 B
主体
控制
准备好
FooA
对象 A
观察者
DOCONNECT
通知

图 36

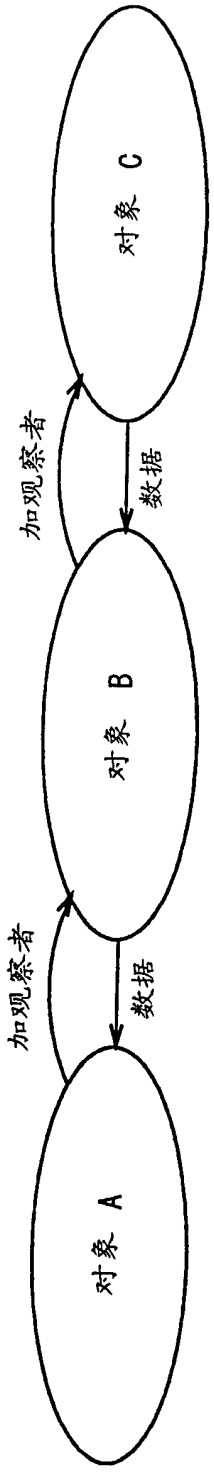


图 37

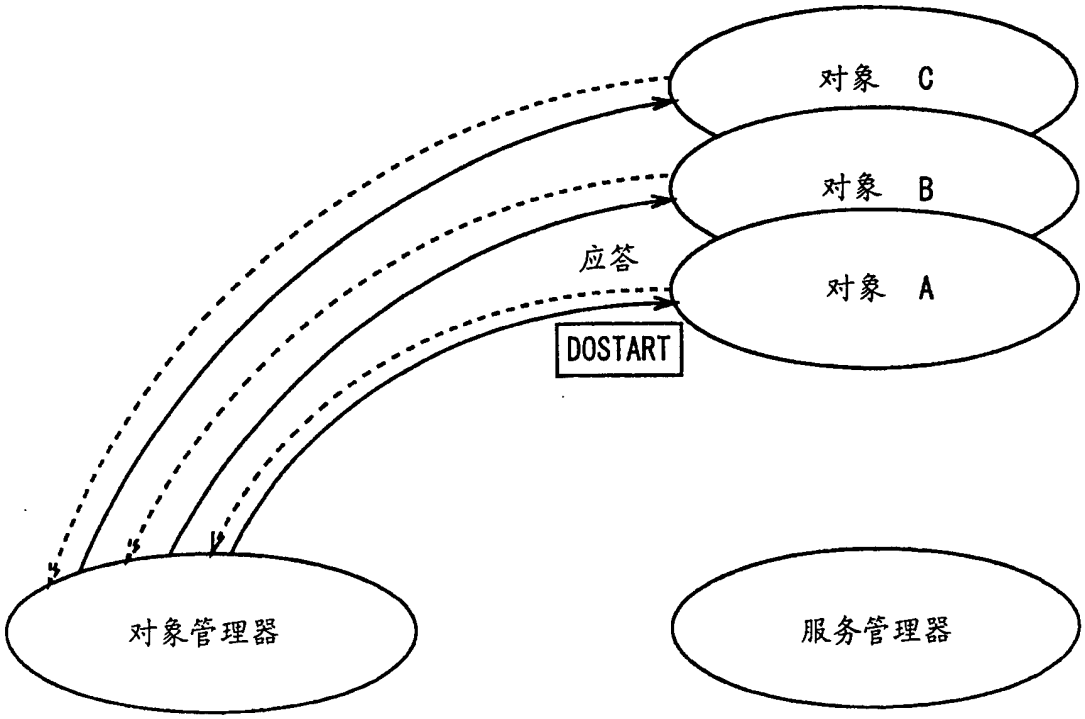


图 38

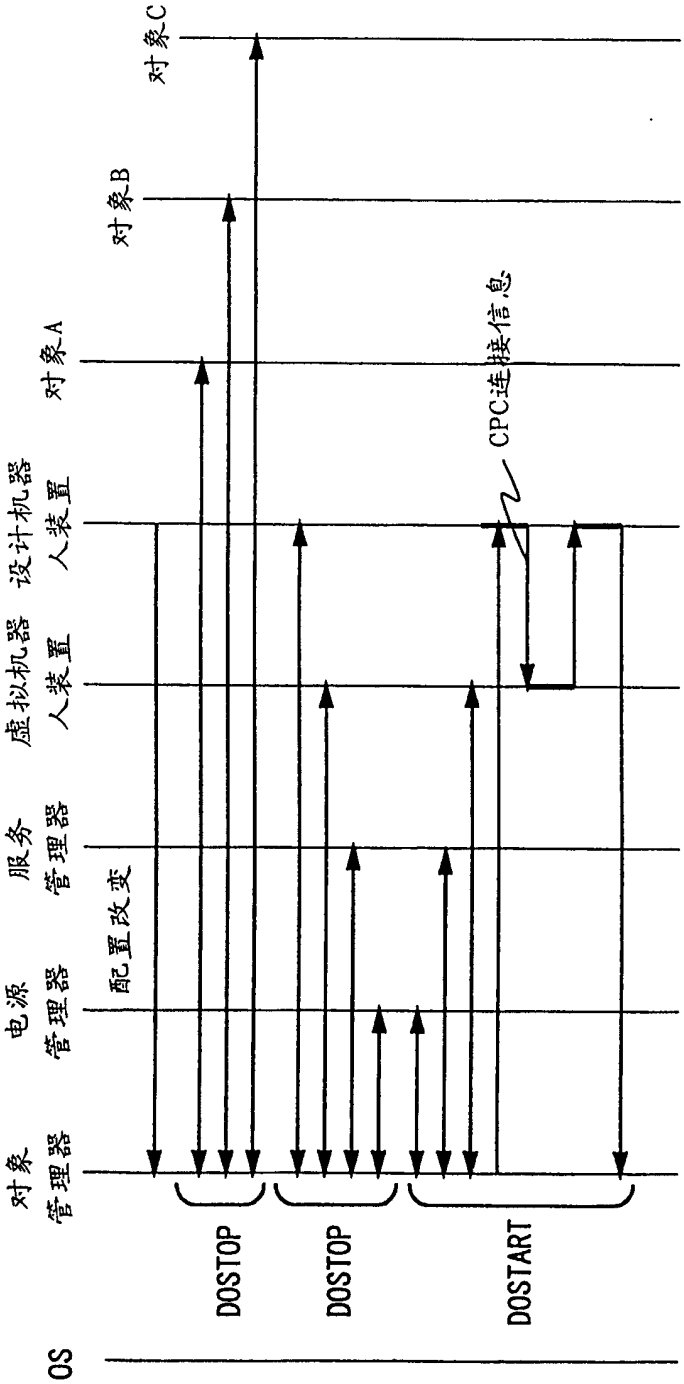

OS
DOSTOP
DOSTOP
DOSTART
对象管理器
电源管理器
服务管理器
虚拟机器人装置
设计机器人装置
对象A
对象B
对象C
配置改变
CPC:连接信息

图 39

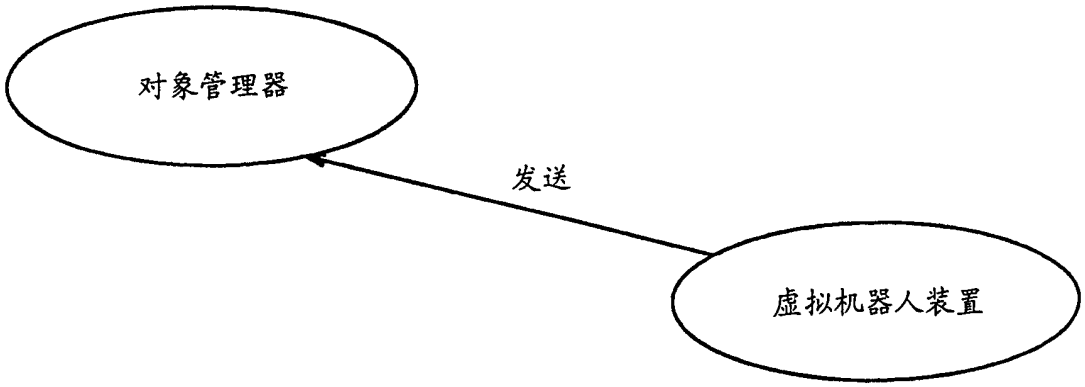


图 40

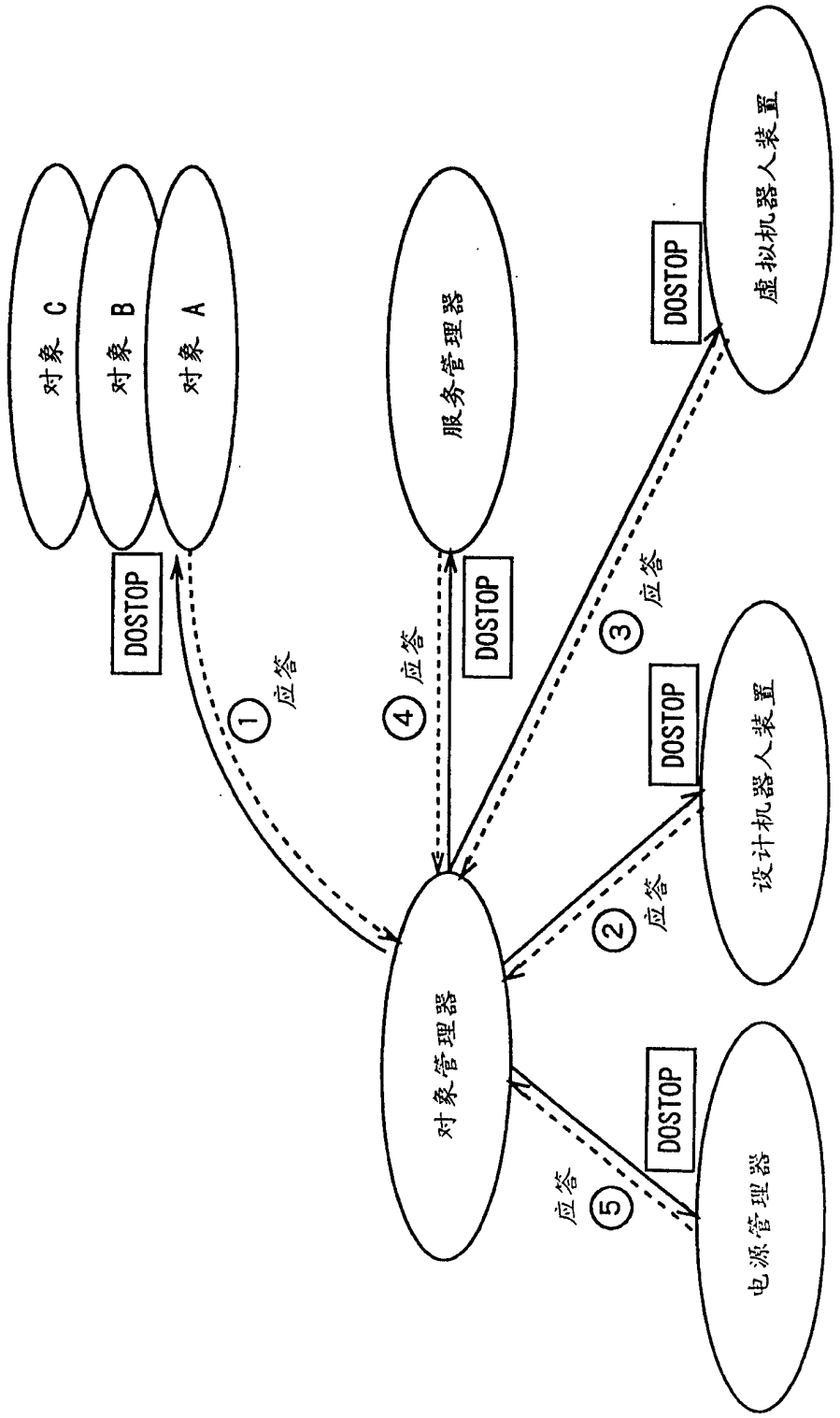

对象 C
对象 B
对象 A
服务管理器
虚拟机器人装置
设计机器人装置
电源管理器
对象管理器
DOSTOP
DOSTOP
DOSTOP
DOSTOP
DOSTOP
① 应答
② 应答
③ 应答
④ 应答
⑤ 应答

图 41

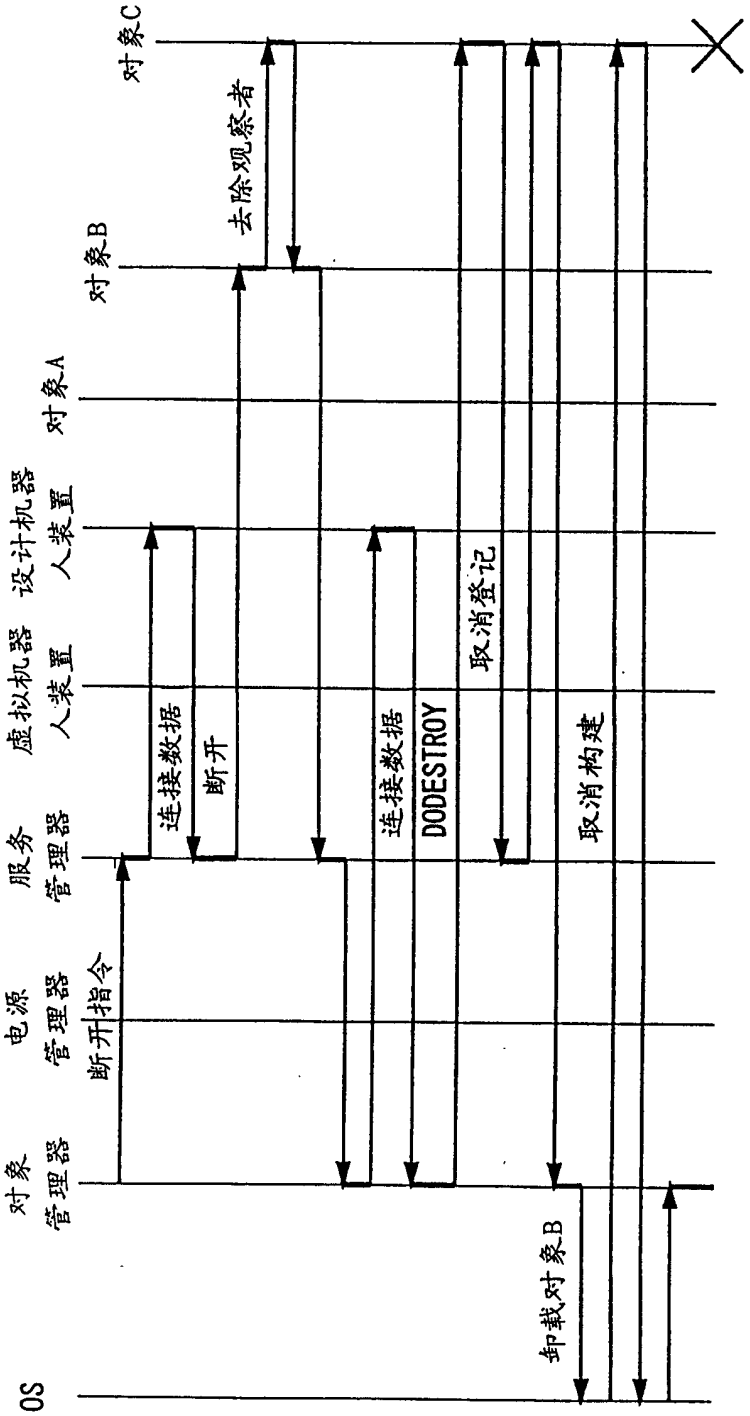

OS
对象管理器
电源管理器
服务管理器
虚拟机器人装置
设计机器人装置
对象A
对象B
对象C
断开指令
连接数据
断开
连接数据
DODESTROY
取消登记
取消构建
去除观察者
卸载对象B

图 42

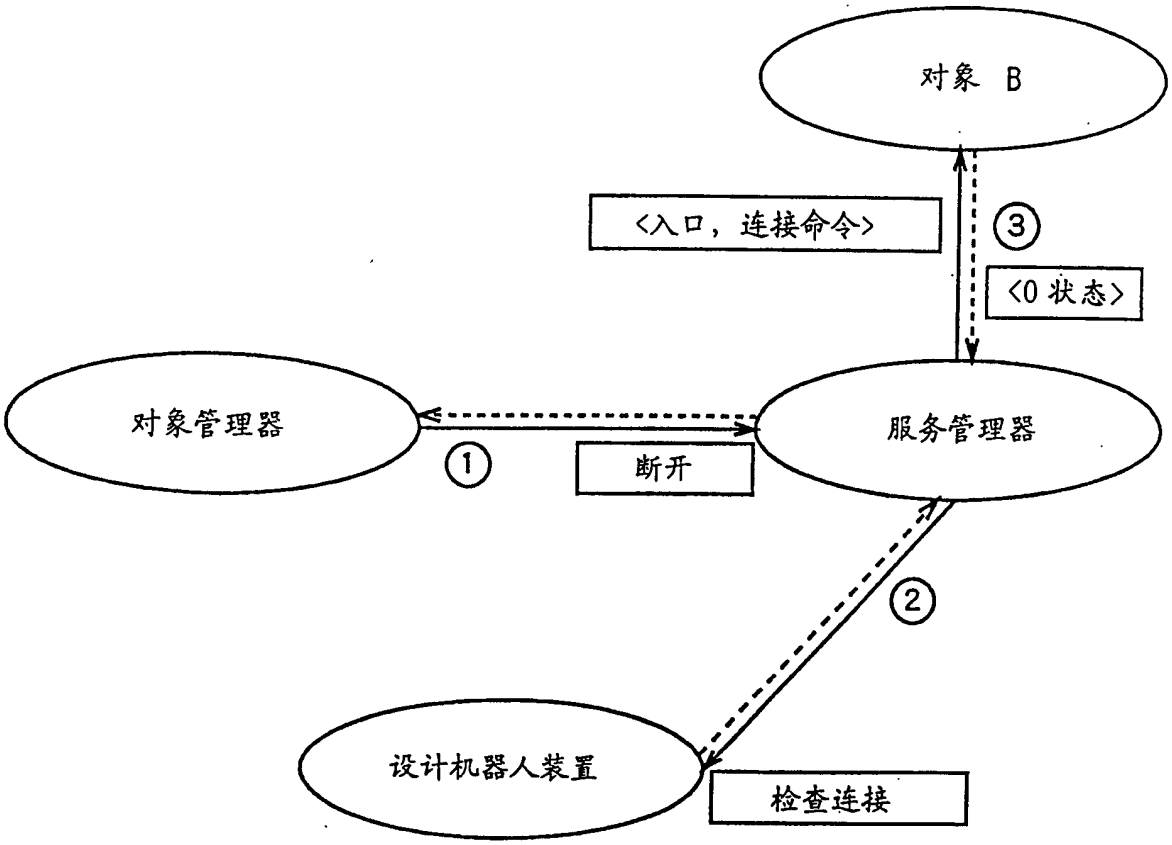


图 43

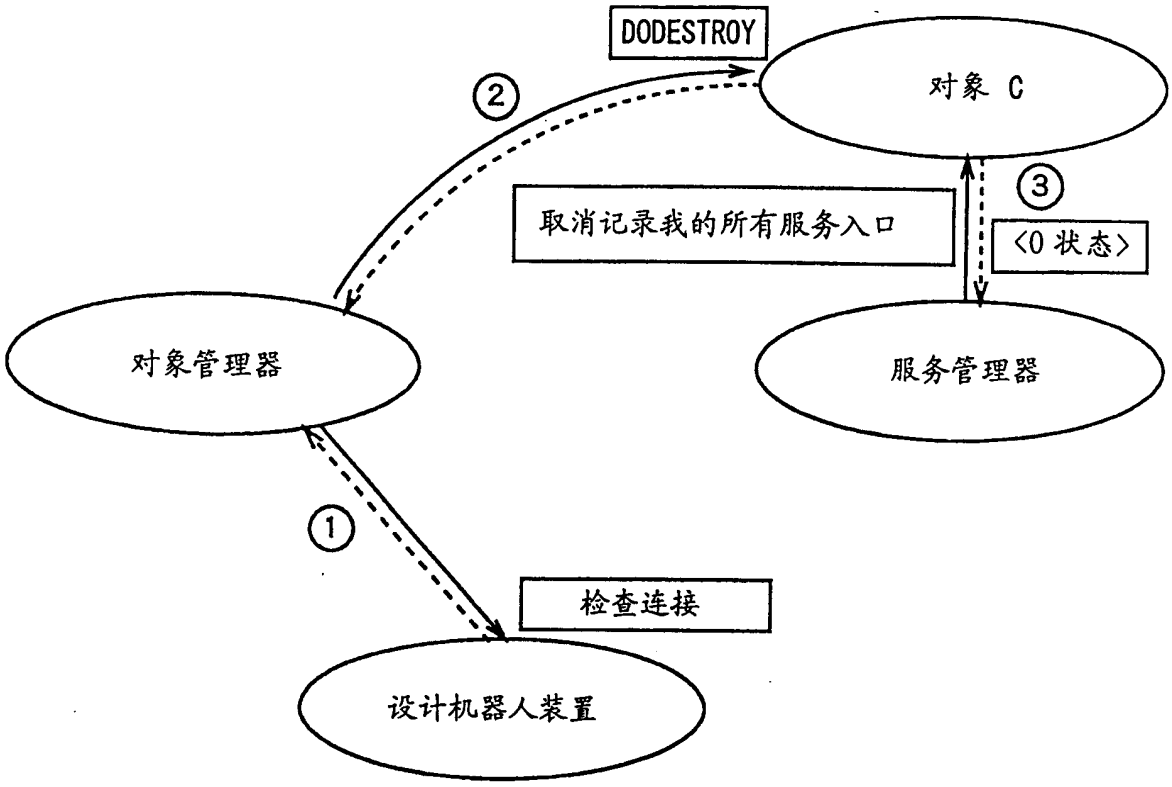

DODESTROY
对象 C
②
③
取消记录我的所有服务入口
〈0 状态〉
对象管理器
服务管理器
①
检查连接
设计机器人装置

图 44

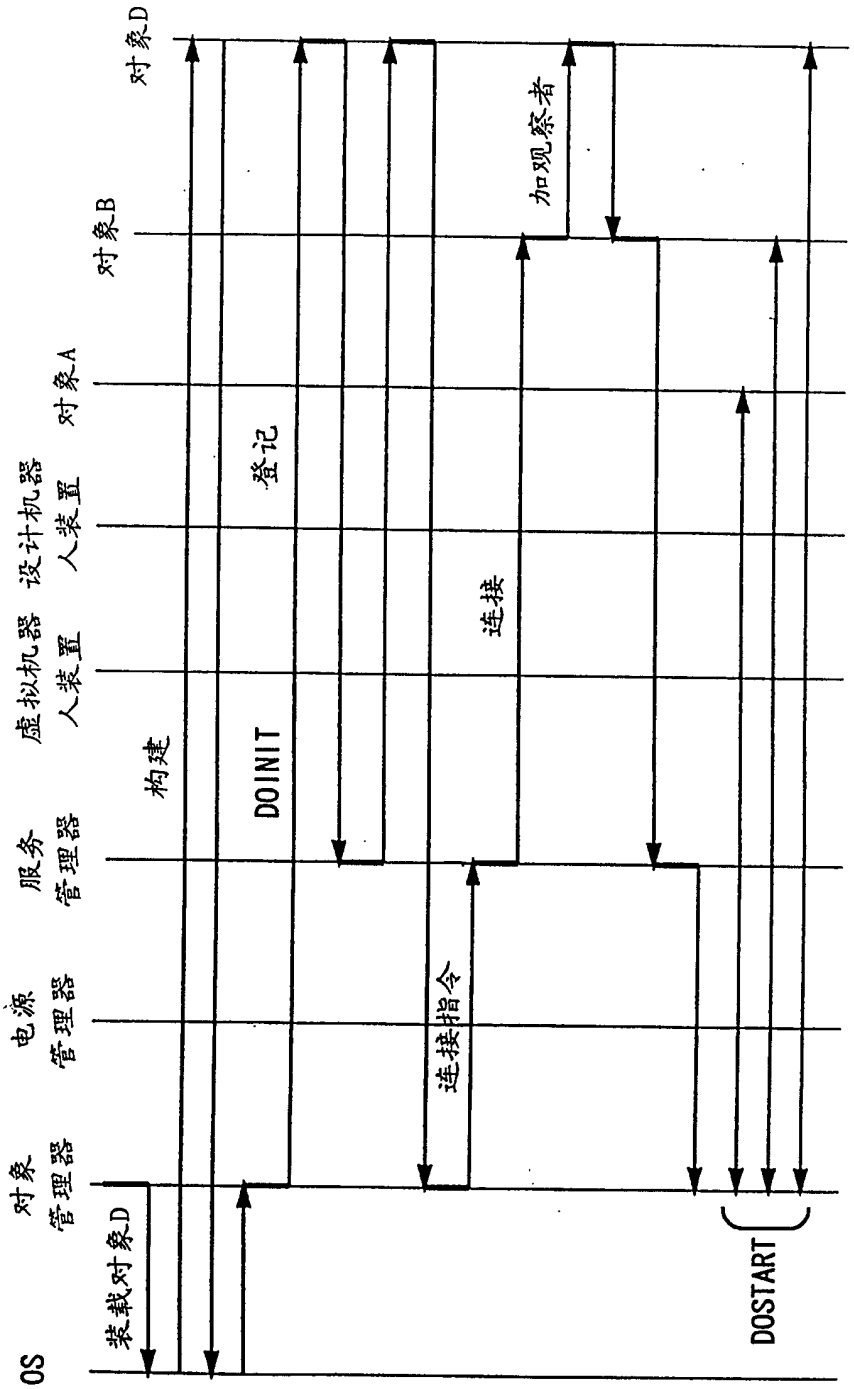

OS
对象管理器
电源管理器
服务管理器
虚拟机器人装置
设计机器人装置
对象A
对象B
对象D
装载对象D
构建
DOINIT
登记
连接指令
连接
加观察者
DOSTART

图 45

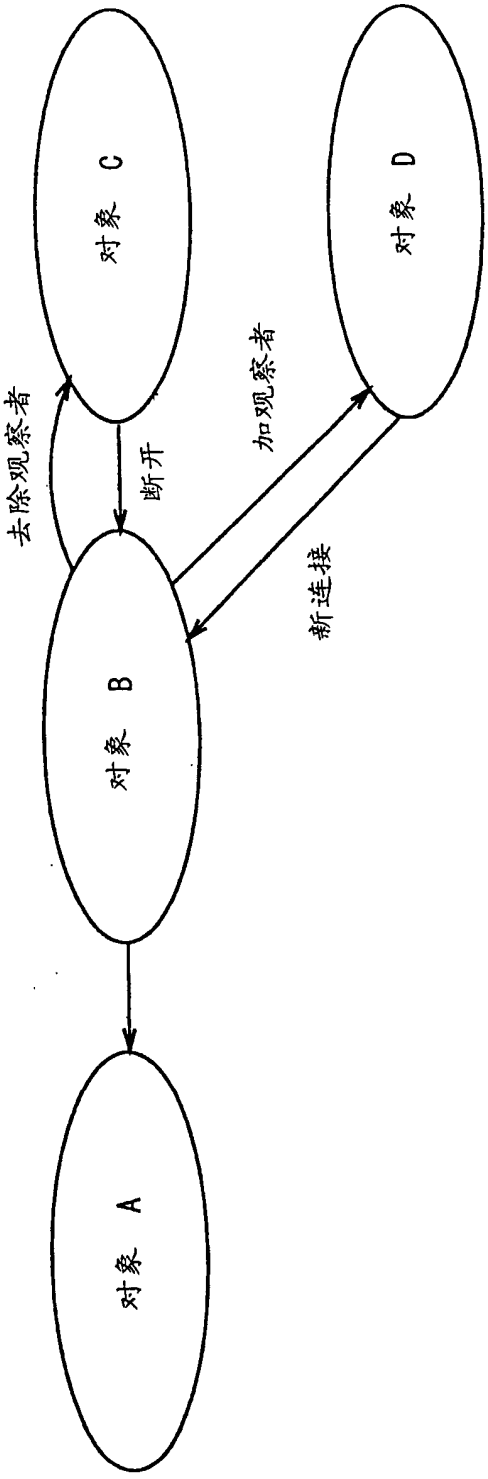


图 46

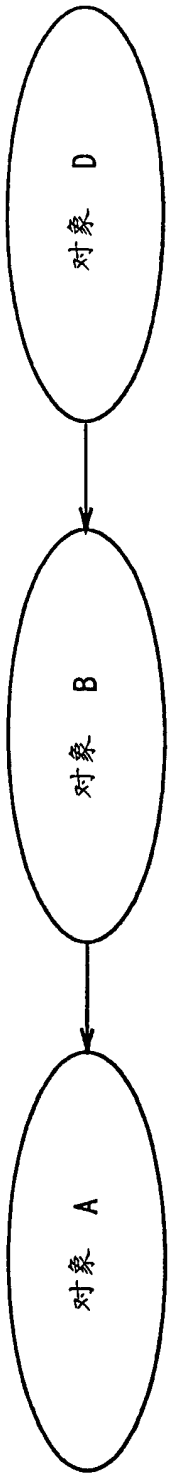


图 47

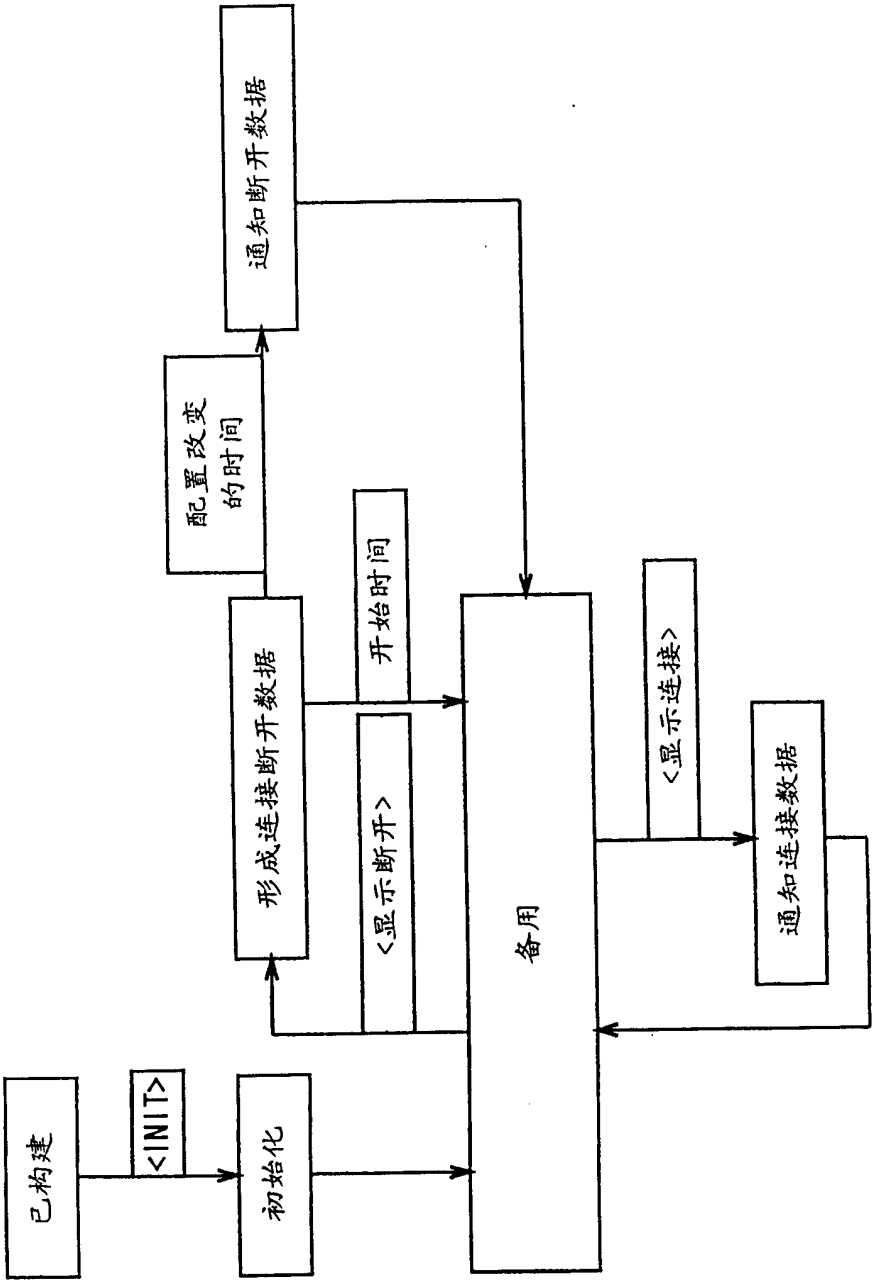

已构建
<INIT>
初始化
形成连接断开数据
<显示断开>
开始时间
配置改变的时间
通知断开数据
备用
<显示连接>
通知连接数据

图 48

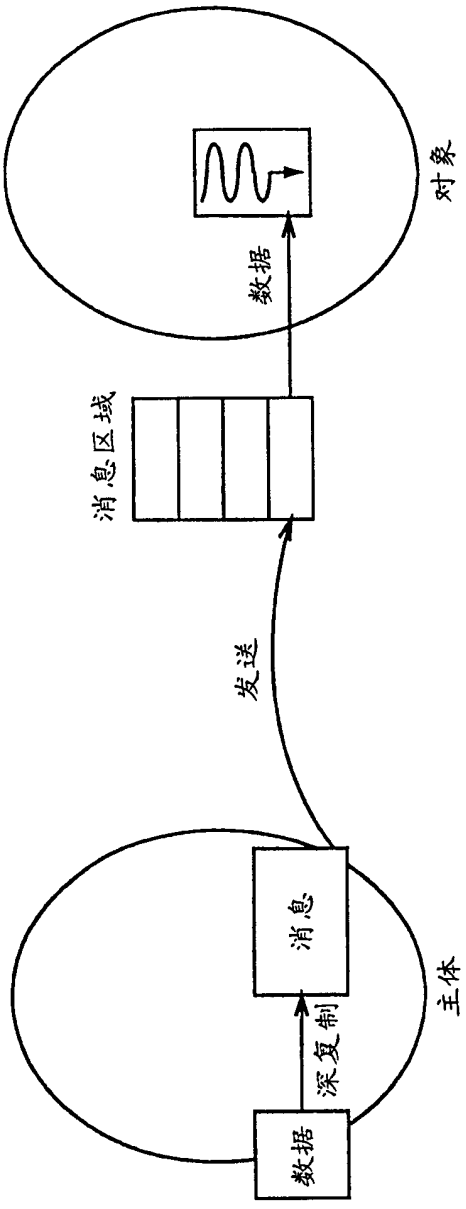


图 49

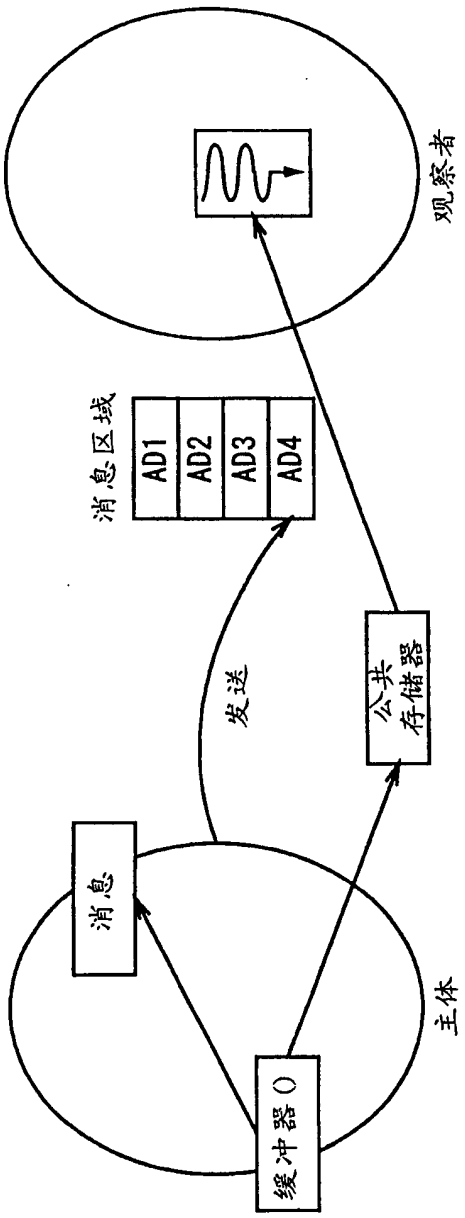


图 50

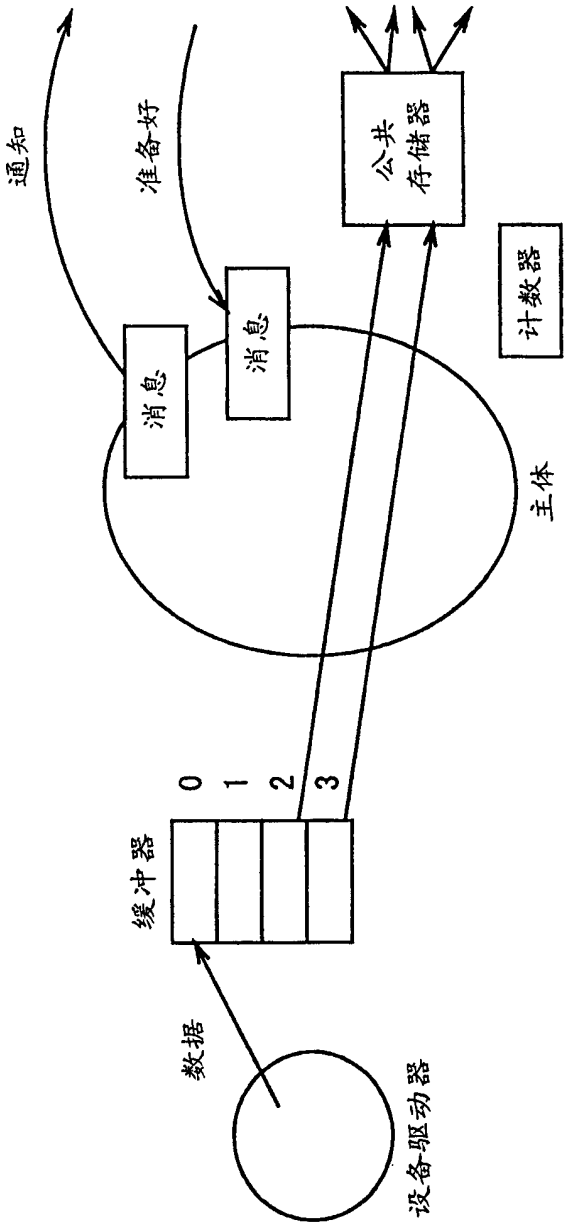


图 51

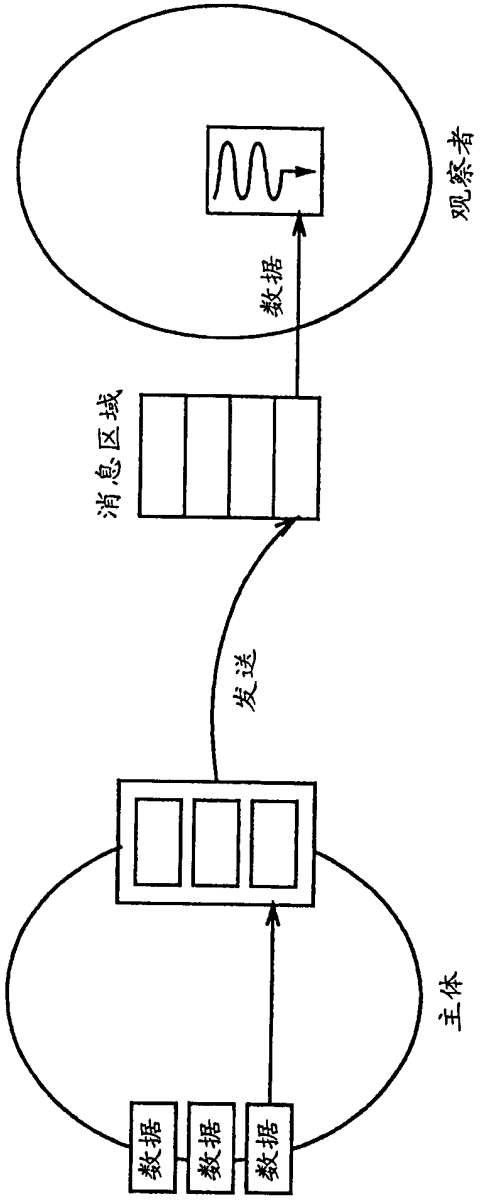


图 52

标号说明

1...机器人装置， 2...主单元， 3...移动单元， 17...电池