



(12) 发明专利申请

(10) 申请公布号 CN 112199686 A

(43) 申请公布日 2021.01.08

(21) 申请号 202011095160.0

(22) 申请日 2008.11.26

(30) 优先权数据

- 60/990,525 2007.11.27 US
- 60/990,442 2007.11.27 US
- 60/990,421 2007.11.27 US
- 60/990,471 2007.11.27 US
- 60/990,492 2007.11.27 US
- 60/990,434 2007.11.27 US
- 60/990,531 2007.11.27 US
- 12/275,651 2008.11.21 US
- 12/276,549 2008.11.24 US
- 12/276,516 2008.11.24 US
- 12/276,728 2008.11.24 US
- 12/277,182 2008.11.24 US
- 12/276,587 2008.11.24 US
- 12/277,174 2008.11.24 US

(62) 分案原申请数据

200880117957.7 2008.11.26

(71) 申请人 波音公司

地址 美国伊利诺伊州

(72) 发明人 W·R·贝克 M·K·罗杰斯

- O·C·沃克 L·福谢斯
- S·W·安格道 J·M·帕克尔
- P·薛 G·A·金伯利
- T·W·古尔德 F·J·麦克来恩
- M·W·哈尼什 T·W·黑德里克逊
- L·H·斯凯福特 C·J·摩根
- K·K·雷恩 D·H·杨

(74) 专利代理机构 北京纪凯知识产权代理有限公司 11245

代理人 李英

(51) Int.Cl.

- G06F 21/57 (2013.01)
- G06F 8/65 (2018.01)
- G06F 9/445 (2018.01)
- G06F 9/44 (2018.01)
- B64F 5/00 (2017.01)
- G06Q 10/08 (2012.01)

权利要求书1页 说明书73页 附图67页

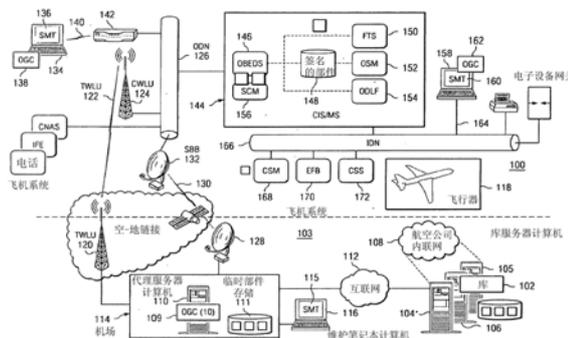
(54) 发明名称

分配可加载软件飞行器部件 (LSAP) 的方法和设备

(57) 摘要

本申请涉及分配可加载软件飞行器部件 (LSAP) 的方法和设备。一种用于管理软件飞行器部件的计算机实现的方法和设备。软件飞行器部件管理设备包括库、接收模块、代理服务器、飞行器上的机载电子分配系统和便携的软件维护工具。地面数据处理系统上的库存储和管理飞行器的软件飞行器部件。接收模块能够从供应商接收软件飞行器部件并且将所述软件飞行器部件发送给库。代理服务器与库通信并且能够将软件分发给多个飞行器客户系统。机载电子分配系统是飞行器客户系统并且能够通过代理服务器从库接收飞行器的软件飞行器部件。便携式软件维护

工具提供将软件飞行器部件发送给飞行器的机载电子分配系统的替代途径。



1. 一种用于处理软件飞行器部件的计算机实现的方法,该计算机实现的方法包括:
从软件飞行器部件管理设备外的源接收包含软件飞行器部件的包;
为输入的包和所述软件飞行器部件验证一组签名;
响应于该组签名有效,将输入的包解包;
显示输入的包的内容;
解包所述包以获取所述软件飞行器部件;
响应于将所述软件飞行器部件上载到所述软件飞行器部件管理设备中的库中的请求,
判断所述软件飞行器部件是否符合规定;
响应于所述软件飞行器部件符合规定,将所述软件飞行器部件签名以形成已签名的软件飞行器部件;
将已签名的软件飞行器部件置于包内,为包含已签名的软件飞行器部件的包签名以形成已签名包,其中已签名的软件飞行器部件和已签名包的签名与输入的包中的该组签名不同;以及
发送已签名包给所述软件飞行器部件管理设备中的库。
2. 根据权利要求1所述的计算机实现的方法,进一步包括:识别所述软件飞行器部件的元数据和其他配置项。
3. 根据权利要求1所述的计算机实现的方法,其中所述软件飞行器部件是第一个签名的软件飞行器部件并且已签名的软件飞行器部件是使用和所述第一个签名的软件飞行器部件不同的签名被签名。
4. 根据权利要求1所述的计算机实现的方法,其中已签名的软件飞行器部件或包的签名来自于制造商、航空公司或供应商。
5. 根据权利要求1所述的计算机实现的方法,其中存储步骤包括:将所述软件飞行器部件存储到文件系统中的目录下。
6. 根据权利要求1所述的计算机实现的方法,进一步包括:显示用户界面,该用户界面能够接收用户的输入从而在多个所存储的软件飞行器部件上进行操作。
7. 一种用于处理配置项的计算机实现的方法,该计算机实现的方法包括:
接收包含配置项的包以形成接收的包;
判断所述包和所述配置项的一组签名是否有效;以及
响应该组签名是有效的判断,保存所述配置项。
8. 根据权利要求7所述的计算机实现的方法,进一步包括判断所述配置项是否有效。
9. 根据权利要求7所述的计算机实现的方法,其中判断所述配置项是否有效的步骤包括:比较所述配置项与规定。
10. 根据权利要求9所述的计算机实现的方法,其中所述配置项是软件飞行器部件并且所述规定是软件飞行器部件标准。

分配可加载软件飞行器部件 (LSAP) 的方法和设备

[0001] 本申请是2008年11月26日提交的名称为“分配可加载软件飞行器部件 (LSAP) 的方法和设备”的中国专利申请200880117957.7的分案申请201710231773.4的分案申请。

技术领域

[0002] 本公开一般涉及改进的数据处理系统并且具体涉及用于管理飞行器的软件的方法和设备。更具体地,本公开涉及计算机实现的方法、设备和计算机可用的程序产品,用于管理可加载的软件飞行器部件和其他关于部件的文件,所述部件被认为是部分使用资源或被简单地认为是资源。

背景技术

[0003] 现代化飞行器及其复杂。例如,飞行器可能具有很多种类型的机载电子系统。飞行器上的一种特定电子系统也可以被称为外场可更换单元 (LRU)。每个外场可更换单元都可以进一步呈现不同的形式。外场可更换单元可以是,例如而不限于飞行管理系统、自动驾驶仪、飞行娱乐系统、通信系统、导航系统、飞行控制器、飞行记录器和碰撞避免系统。外场可更换单元可以使用软件或程序,从而为不同的操作和功能提供逻辑或控制。在这些外场可更换单元中使用的软件通常被视为航空业的一部分。具体地,使用在飞行器上的外场可更换单元中的软件应用也可以被单独地跟踪并且被称为可加载软件飞行器部件 (LSAP) 或软件飞行器部件。如美国联邦航空局 (FAA) 所定义的,该软件应用程序也可以被认为是飞行器配置的一部分。

[0004] 当某个实体 (例如航空公司、维护、修理和检修服务供应商“MRO”或军事飞行中队) 收到飞行器时,软件飞行器部件典型地已经被安装在飞行器中的外场可更换单元中。例如,航空公司在这类软件飞行器部件需要被重新安装或重新加载到已经故障和已经被替换的外场可更换单元的情况下也可以收到这些软件飞行器部件的复本。进一步地,航空公司有时也可以接收可加载软件飞行器部件的更新。这些更新可以包括在当前所安装的软件飞行器部件中没有的额外特征,并且可以被认为是对一个或更多外场可更换单元的升级。

[0005] 用于管理、处理和分配可加载软件飞行器部件的当前系统是繁琐的并且是耗时的。当前,软件飞行器部件被存储在物理介质中,例如磁盘、光盘或数字多功能光盘 (DVD)。航空公司收到传送的物理介质并且将该物理介质保存在诸如档案柜的地方。在许多情况下,该介质也可以被放置在飞行器上。

[0006] 有时也可以对飞行器进行维护操作以安装或重新安装软件飞行器部件。当需要一个软件飞行器部件时,必须定位和获取包含该部件的介质以供维护人员使用。这种类型存储和获取系统和过程占用空间和时间。

[0007] 因此,具有改进的方法和设备用于分配软件飞行器部件,以解决上述问题是有利的。

发明内容

[0008] 不同的有利实施例提供计算机实现的方法和设备,用于管理软件飞行器部件。在一个有利实施例中,软件飞行器部件管理设备包括库、接收模块、代理服务器、飞行器上的机载电子分配系统和便携式的软件维护工具。地面数据处理系统上的库为多个飞行器存储和管理多个软件飞行器部件。接收模块能够从供应方接受该多个软件飞行器部件并且将该多个软件飞行器部件发送给库进行存储。代理服务器与库进行通信并且能够将软件分配给多个飞行器客户系统。机载电子分配系统是飞行器客户系统并且能够通过代理服务器从库为飞行器接收软件飞行器部件。便携式软件维护工具提供将软件飞行器部件发送到飞行器的机载电子分配系统的替代途径。

[0009] 在另一个有利实施例中,一种设备包括库、代理服务器和机载电子分配系统。库位于数据处理系统上并且存储和管理飞行器部件。代理服务器能够从库接收软件飞行器部件以分配给一组飞行器。飞行器的机载电子分配系统能够将代理服务器接收的软件飞行器部件分配到飞行器中的外场可更换单元。

[0010] 在再一个有利实施例中,计算机实现的方法用于管理多个软件飞行器部件。软件飞行器部件保存在库中。为软件飞行器部件识别飞行器以形成已识别的飞行器。软件飞行器部件被发送到代理服务器。软件飞行器部件从代理服务器被发送到已识别飞行器上的机载电子分配系统。

[0011] 有利实施例提供一种计算机实现的方法、设备和计算机程序产品,用于处理软件飞行器部件。在一个有利实施例中,一种方法用于处理软件飞行器部件。从软件飞行器部件管理设备外的源处接收包含软件飞行器部件的输入包。为该输入包和软件飞行器部件验证一组签名。响应该组签名是有效的,解包(打开)输入包。输入包的内容被显示。软件飞行器部件被解包以获取软件飞行器部件。响应将保存的软件飞行器部件上载到软件飞行器部件管理设备中的库的请求,判断所保存的软件飞行器部件是否符合规定。如果所保存的软件飞行器部件符合规定,那么所保存的软件飞行器部件被签名以形成已签名的软件飞行器部件。已签名的软件飞行器部件被置入输入包。包含已签名的软件飞行器部件的包被签名以形成已签名的包,其中已签名的软件飞行器部件和已签名的包的签名与输入包中的该组签名不同。已签名的包被发送到软件飞行器部件管理设备中的库。

[0012] 在另一个有利实施例中,计算机实现的方法用于处理配置项。接收包含配置项的包以形成接收的包。判断包和配置项的一组签名是否是有效的。响应该组签名被判断为有效而存储配置项。

[0013] 在再一个有利实施例中,数据处理系统包括总线、存储装置和处理器单元。存储装置连接到总线并且包括程序代码。处理器单元连接到总线并且能够执行所述程序代码以从软件飞行器部件管理设备外的源处接收包含软件飞行器部件的包,验证输入包和软件飞行器部件的一组签名,响应该组签名是有效的而解包输入包,显示输入包的内容,存储软件飞行器部件,响应将软件飞行器部件上载到软件飞行器部件管理设备的库中的请求,判断软件飞行器部件是否符合规定、响应符合规定的软件飞行器部件而将软件飞行器部件签名以形成已签名的软件飞行器部件,将已签名的软件飞行器部件置于包内,将包含已签名的软件飞行器部件签名以形成已签名的包,其中已签名的软件飞行器部件和已签名的包的签名与输入包中的该组签名不同,以及将已签名的包发送到软件飞行器部件管理设备中的库。

[0014] 不同的有利实施例提供计算机实现的方法、设备和计算机程序产品,用于管理软件飞行器部件。在一个有利的实施例中,方法包括从飞行器网络数据处理系统中的库检索一组软件飞行器部件和关于该组软件飞行器部件的元数据,以形成一组检索到的软件飞行器部件。该组软件飞行器部件被存储在文件系统中。元数据被存储在数据库中。该组软件飞行器部件和元数据被发送到机载电子分配系统以及地面组件接口和机载电子分配系统之间的通信链接。关于通信链接上的活跃性的状态信息被保持。

[0015] 在另一个有利实施例中,设备包括:文件系统、数据库、存储在文件系统中的一组软件飞行器部件、存储在数据库中的一组命令、地面组件、控制进程和数据处理系统。地面组件能够与在多个飞行器上的多个机载电子分配系统交换信息。控制进程能够从库中接收该组命令和该组软件飞行器部件,并且将该组命令和该组软件飞行器部件发送到多个飞行器。文件系统、数据库、该组软件飞行器部件、该组命令、地面组件和控制进程是数据处理系统中的软件组件。

[0016] 在又一个有利实施例中,计算机程序产品包括计算机可读介质和存储在该计算机可读介质上的程序代码。给出程序代码,用于从飞行器网络数据处理系统中的库检索一组软件飞行器部件和关于该组软件飞行器部件的元数据以形成一组检索到的软件飞行器部件。程序代码被存储在计算机可读介质中用于在文件系统中存储该组软件飞行器部件。也给出用于将元数据存储在数据库中的程序代码。程序代码被存储在计算机可读介质中用于将该组软件飞行器部件和元数据发送到机载电子分配系统以及地面组件接口和机载电子分配系统之间的通信链接。给出程序代码用于维持关于通信链接的活跃性的状态信息。

[0017] 不同的有利实施例提供计算机实现的方法、设备和计算机程序产品,用于处理网络数据处理系统中的命令。在一个有利的实施例中,一进程为多个飞行器中的飞行器创建一组命令。该命令组被存储在网络数据处理系统中的库中的一组队列中,以形成存储的命令组。响应事件,存储的命令组被发送到在网络数据处理系统上执行的代理服务器应用程序以形成已分配的命令组。建立代理服务器应用程序与在飞行器上的飞行器数据处理系统上执行的机载分配系统之间的连接。机载电子分配系统从代理服务器应用程序请求已分配的命令组。机载电子分配系统从代理服务器应用程序接收已分配的命令组以形成接收的命令组。该接收的命令组由机载电子分配系统处理,以形成经处理的命令组。如果经处理的命令组是一组上行链路命令,则使用该组经处理的命令将一组软件飞行器部件转发到在飞行器上的一组目标数据处理系统。

[0018] 在另一个有利实施例中,计算机实现的方法识别在网络数据处理系统中的库中的一组命令,以形成已识别的命令组。该已识别的命令组被发送到在网络数据处理系统上执行的代理服务器应用程序以形成已分配的命令组。在代理服务器应用程序与在飞行器上的飞行器数据处理系统上执行的机载电子分配系统之间建立连接。从代理服务器应用程序发送已分配的命令组到机载电子分配系统以形成通过连接接收的命令组。接收的命令组由机载电子分配系统处理以形成已处理的命令组。如果已处理的命令组是一组上行链路命令,那么使用已处理的命令组将位于飞行器数据处理系统中的存储的软件飞行器部件组发送到飞行器上的一组目标数据处理系统。

[0019] 在又一个有利实施例中,计算机程序产品包括计算机可读介质和存储在该计算机可读介质中的程序代码。程序代码包括用于为在多个飞行器中的飞行器创建一组命令的程

序代码和用于将该组命令存储在网络数据处理系统中的库中的一组队列中以形成存储的命令组的程序代码。程序代码也包括用于响应事件,将存储的命令组发送到在网络数据处理系统上执行的代理服务器应用程序以形成已分配的命令组的程序代码。程序代码也包括建立代理服务器应用程序和在飞行器上的飞行器数据处理系统上执行的机载电子分配系统之间的连接的程序代码,和用于由机载电子分配系统从代理服务器应用程序请求已分配的命令组的程序代码。程序代码包括用于由机载电子分配系统从代理服务器应用程序接收分配的命令组以形成接收的命令组的程序代码和用于由机载电子分配系统处理接收的命令组以形成已处理的命令组的程序代码。该程序代码也包括程序代码,用于在如果已处理的命令组是一组上行链路命令时使用已处理的命令组将一组软件飞行器部件转发给飞行器上的一组目标数据处理系统。

[0020] 不同的有利实施例提供计算机实现的方法、设备和计算机程序产品,用于管理软件飞行器部件。在一个有利实施例中,一种方法被用于通过地面网络建立便携式数据处理系统和源之间的连接以形成建立的连接。通过建立的连接从源检索一组上行链路命令。通过建立的连接从源检索与该组上行链路命令相对应的一组软件飞行器部件以形成一组检索到的软件飞行器部件。在便携式数据处理系统中存储该组检索到的软件飞行器部件以形成一组存储的软件飞行器部件。断开具有该组存储的软件飞行器部件的便携式数据处理系统与地面网络的连接。将具有该组存储的软件飞行器部件的便携式数据处理系统连接到飞行器中的飞行器网络。通过便携式数据处理系统中的地面组件从该组上行链路命令发出上行链路命令给飞行器网络。通过地面组件发送与该上行链路命令相对应的存储的软件飞行器部件给飞行器网络。

[0021] 在另一个有利实施例中,设备包括:数据库、文件系统、库服务、管理器和便携式数据处理系统。数据库能够存储一组命令并且文件系统能够存储一组软件飞行器部件。库服务能够与从库和代理服务器应用程序之一中选出的源进行通信,地面组件能够与飞行器网络中的飞行器数据处理系统直接连接。管理器能够通过库服务从地面网络中的源接收该组命令和该组软件飞行器部件,断开和源的连接,并且通过地面组件发送该组命令和该组软件飞行器部件给飞行器网络中的飞行器数据处理系统。便携式数据处理系统能够建立与地面网络和飞行器网络的连接,其中库服务、地面组件、数据库和文件系统位于便携式数据处理系统上。

[0022] 在还另一个有利实施例中,计算机程序产品包括计算机可记录存储介质和存储在计算机可记录存储介质上的程序代码。该程序代码包括用于通过地面网络建立便携式数据处理系统和源之间的连接以形成建立的连接的代码。该程序代码也包括代码,用于通过建立的连接从源检索一组上行链路命令并且通过建立的连接从源检索与该上行链路命令相对应的一组软件飞行器部件以形成一组检索到的软件飞行器部件。该程序代码包括代码,用于在便携式数据处理系统中存储该组检索到的软件飞行器部件以形成一组存储的软件飞行器部件。该程序代码具有代码,用于断开具有该组存储的软件飞行器部件的便携式数据处理系统与地面网络的连接并且用于将具有该组存储的软件飞行器部件的便携式数据处理系统连接到飞行器中的飞行器网络。该程序代码具有代码,用于通过便携式数据处理系统中的地面组件从该组上行链路命令发出上行链路命令给飞行器网络并且用于通过地面组件发送与该上行链路命令相对应的存储软件飞行器部件给飞行器网络。

[0023] 不同的有利实施例提供计算机实现的方法、设备和计算机程序产品,用于与飞行器传输信息。在一个有利实施例中,计算机实现的方法被用于与飞行器传输信息。在执行在飞行器中的飞行器数据处理系统中的机载电子分配系统和地面组件之间建立连接。响应地面组件通过连接做出的对命令的请求,由机载电子分配系统执行的命令被识别,以形成已识别的命令。该已识别的命令从地面组件发送给机载电子分配系统。将事务处理标识符分配给命令。与该命令关联的事务处理状态通过使用事物处理标识符被保持在机载电子分配系统和地面组件上。由地面组件启动上行链路。软件飞行器部件被从地面组件发送给机载电子分配系统以进行上传。地面组件上的软件飞行器部件的传输状态被保存。

[0024] 在另一个有利实施例中,计算机实现的方法被用于与飞行器传输信息。从地面组件请求命令。响应从地面组件收到上行链路命令,请求与该上行链路命令相对应的软件飞行器部件。响应发送对软件飞行器部件的请求,从地面组件接收软件飞行器部件以形成接收的软件飞行器部件。该软件飞行器部件被存储。

[0025] 在又一个有利实施例中,设备包括:地面组件、机载电子分配系统、数据处理系统和飞行器数据处理系统。机载电子分配系统能够从地面组件接收命令;响应从地面组件接收上行链路命令而请求对应于上行链路命令的软件飞行器部件;响应发送对软件飞行器部件的请求,从地面组件接收软件飞行器部件以形成接收的软件飞行器部件;以及存储软件飞行器部件。地面组件在数据处理系统上执行。机载电子分配系统在飞行器数据处理系统上执行。

[0026] 不同的有利实施例提供计算机实现的方法、设备和计算机程序产品,用于管理软件飞行器部件。在一个有利实施例中,在库接收软件飞行器部件以形成接收的软件飞行器部件。将接收的软件飞行器部件存储在具有多个软件飞行器部件的库中的存储器中。提供用户界面系统以操控库中的多个软件飞行器部件。响应从用户界面系统接收从库中的多个软件飞行器部件上传所选的软件飞行器部件给目标飞行器数据处理系统的用户输入,创建命令以发送所选的软件飞行器部件给在其中置有目标飞行器数据处理系统的飞行器以形成上行链路命令。将上行链路命令存储到库中的存储器中的队列中以形成存储的上行链路命令。响应来自代理服务器应用程序的对命令的请求,分配存储的上行链路命令给代理服务器应用程序。响应接收来自处理所存储的上行链路命令的代理服务器应用程序的对所选软件飞行器部件的请求,发送所选的软件飞行器部件给代理服务器应用程序。

[0027] 在另一个有利实施例中,设备包括位于存储装置上的文件系统、位于存储装置上的数据库、管理组件和用户界面。文件系统存储多个软件飞行器部件并且数据库存储用于管理多个软件飞行器部件的多个命令。管理组件能够生成多个命令。用户界面系统能够接收用户输入以操作管理组件。

[0028] 在又一个有利实施例中,计算机包括:总线、连接到总线的通信适配器、连接到总线的存储器以及连接到总线的处理器单元,其中计算机可用的程序代码被存储在总线上。处理器单元执行计算机可用程序代码以在库上接收软件飞行器部件,从而形成接收的软件飞行器部件;在具有多个软件飞行器部件的库中的存储器中存储接收的软件飞行器部件;提供用户界面系统以操控库中的多个软件飞行器部件;响应从用户界面系统接收将从库的多个软件飞行器部件中选出的软件飞行器部件上传给目标飞行器数据处理系统的用户输入,创建命令以发送所选的软件飞行器部件给在其中置有目标飞行器数据处理系统的飞行

器,从而形成上行链路命令;将上行链路命令存储到库中的存储器中的队列中以形成存储的上行链路命令;响应来自代理服务器应用程序的对命令的请求,分配存储的上行链路命令给代理服务器应用程序;以及响应从处理所存储的上行链路命令的代理服务器应用程序接收的对所选软件飞行器部件的请求,发送所选的软件飞行器部件给代理服务器应用程序。

[0029] 在再另一个有利实施例中,计算机程序产品包括:计算机可记录的存储介质和程序代码。该程序代码包括用于接收库上的软件飞行器部件以形成接收的软件飞行器部件的代码。程序代码也包括用于将接收的软件飞行器部件存储在具有多个软件飞行器部件的库中的存储器中以及提供用户界面系统以操控库中的多个软件飞行器部件的代码。程序代码包括代码,用于响应从用户界面系统接收从库中的多个软件飞行器部件上传所选的软件飞行器部件给目标飞行器数据处理系统的用户输入,创建命令以发送所选的软件飞行器部件给在其中置有目标飞行器数据处理系统的飞行器以形成上行链路命令。也给出代码,用于将上行链路命令存储到库中的存储器中的队列中以形成存储的上行链路命令。程序代码包括用于响应来自代理服务器应用程序的对命令的请求,分配存储的上行链路命令给代理服务器应用程序的代码,以及用于响应接收来自处理所存储的上行链路命令的代理服务器应用程序的对所选软件飞行器部件的请求,发送所选的软件飞行器部件给代理服务器应用程序的代码。

[0030] 特征、功能和优势可以独立地在本公开的多个实施例中获得或者也可以与其他实施例结合,在这些实施例中的进一步的细节可以参考下列描述和附图看出。

附图说明

[0031] 在所附的权利要求中阐述被认为是新颖特征的有利实施例的特性。然而,参考下列对本发明的有利实施例的详细描述,结合附图阅读将最佳地理解有利实施例以及优选的使用方式及其目的和优点。

[0032] 图1是根据说明性实施例的数据处理环境的示例性图示;

[0033] 图2是根据说明性实施例的数据处理系统的图示;

[0034] 图3是图解说明根据有利实施例的软件飞行器部件管理设备的图示;

[0035] 图4是根据有利实施例的软件部件管理环境的框图;

[0036] 图5是图解说明根据有利实施例的软件部件管理环境的操作模式的表格;

[0037] 图6是图解说明根据有利实施例的命令类型的图示;

[0038] 图7是根据有利实施例的命令格式;

[0039] 图8是图解说明根据有利实施例处理上行链路命令的消息流图示;

[0040] 图9是图解说明根据有利实施例处理下行链路命令的消息传送图示;

[0041] 图10是图解说明根据有利实施例的删除命令的处理的消息流图示;

[0042] 图11是根据有利实施例,用于分配软件飞行器部件的过程的高级流程图;

[0043] 图12是根据有利实施例,用于接收和存储软件飞行器部件的过程流程图;

[0044] 图13是根据有利实施例,通过代理服务器分配命令的过程流程图;

[0045] 图14是根据有利实施例,通过代理服务器应用程序接收和分配下行链路数据的过程流程图;

- [0046] 图15是根据有利实施例,使用软件维护工具分配软件飞行器部件的过程流程图;
- [0047] 图16是根据有利实施例,使用软件维护工具接收数据的过程流程图;
- [0048] 图17是根据有利实施例的库的功能框图;
- [0049] 图18是图解说明根据有利实施例的文件系统目录布局的图示;
- [0050] 图19是图解说明根据有利实施例的队列中命令组织的框图;
- [0051] 图20是根据有利实施例的软件飞行器部件的框图;
- [0052] 图21是根据有利实施例的删除命令的命令数据结构;
- [0053] 图22是图解说明根据有利实施例的上行链路命令的命令数据结构的图示;
- [0054] 图23是图解说明根据有利实施例的下行链路命令的命令数据结构的图示;
- [0055] 图24是根据有利实施例的调度命令的用户界面的图示;
- [0056] 图25是图解说明根据有利实施例,用于浏览命令的用户界面的图示;
- [0057] 图26是根据有利实施例,用于浏览部件的用户界面的图示;
- [0058] 图27是根据有利实施例,用于在库中接收软件飞行器部件的过程流程图;
- [0059] 图28是根据有利实施例,用于创建命令的过程流程图;
- [0060] 图29是根据有利实施例管理软件飞行器部件的过程的高级流程图;
- [0061] 图30是根据有利实施例调度命令结构的过程流程图;
- [0062] 图31是根据有利实施例调度命令文件的过程流程图;
- [0063] 图32是根据有利实施例调度部件的过程流程图;
- [0064] 图33是根据有利实施例的出列命令的过程流程图;
- [0065] 图34是图解说明根据有利实施例的代理服务器应用程序中的数据流的图示;
- [0066] 图35是图解说明根据有利实施例的代理服务器应用程序的图示;
- [0067] 图36-39是图解说明根据有利实施例的数据结构的图示;
- [0068] 图40是根据有利实施例的代理服务器文件系统目录结构的图示;
- [0069] 图41是根据有利实施例从库中接收信息的过程流程图;
- [0070] 图42是根据有利实施例发送下行链路文件给库的过程流程图;
- [0071] 图43是根据有利实施例库发送事件文件给库的过程流程图;
- [0072] 图44是根据有利实施例发送信息给飞行器的过程流程图;
- [0073] 图45是根据有利实施例接收软件飞行器部件的过程流程图;
- [0074] 图46是根据有利实施例从飞行器接收命令状态信息的过程流程图;
- [0075] 图47是根据有利实施例接收下行链路文件的过程流程图;
- [0076] 图48是根据有利实施例从软件维护工具接收状态信息的过程流程图;
- [0077] 图49是根据有利实施例向软件维护工具发送信息的过程流程图;
- [0078] 图50是根据有利实施例发送软件飞行器部件的列表给软件维护工具的过程流程图;
- [0079] 图51是根据有利实施例从软件维护工具接收下行链路文件的过程流程图;
- [0080] 图52是根据有利实施例从软件维护工具接收事件日志文件的过程流程图;
- [0081] 图53是图解说明根据有利实施例的数据流和软件维护工具的图示;
- [0082] 图54是根据有利实施例的软件维护工具的框图;
- [0083] 图55是根据有利实施例描写的命令表和命令资源表的图示;

- [0084] 图56是根据有利实施例的部分下行链路数据的图示；
- [0085] 图57是根据有利实施例的下行链路表的图示；
- [0086] 图58是根据有利实施例的软件维护工具文件系统目录结构的图示；
- [0087] 图59是图解说明根据本发明有利实施例的实现在软件维护工具中的界面组件的图示；
- [0088] 图60-65是根据本发明有利实施例的用户界面组件的用户界面的示例性实现；
- [0089] 图66是图解说明根据本发明有利实施例,通过软件维护工具发送命令和软件飞行器部件给飞行器的数据流的图示；
- [0090] 图67是图解说明根据本发明有利实施例在处理下传的文件的过程中在软件维护工具中的数据流的图示；
- [0091] 图68是图解说明根据本发明有利实施例的数据流和由软件维护工具记录输入事件的图示；
- [0092] 图69是图解说明根据本发明有利实施例从库检索部件过程中软件维护工具中的数据流的图示；
- [0093] 图70是图解说明根据本发明有利实施例在检索和创建命令期间软件维护工具中的数据流的图示；
- [0094] 图71是图解说明根据本发明有利实施例从可替代源上传软件飞行器部件的图示；
- [0095] 图72是根据本发明有利实施例用于管理软件飞行器部件的过程的高级流程图；
- [0096] 图73是根据本发明有利实施例用于管理软件飞行器部件的过程的更详细流程图；
- [0097] 图74是根据本发明有利实施例从软件维护工具发送软件飞行器部件给机载电子分配系统的过程流程图；
- [0098] 图75是根据本发明有利实施例接收下行链路数据的过程流程图；
- [0099] 图76是根据本发明有利实施例被用于与飞行器传输信息的组件的图示；
- [0100] 图77是图解说明根据本发明有利实施例被用于轮询命令的消息流的消息流图示；
- [0101] 图78-79是图解说明根据本发明有利实施例发送状态信息的信息流图示；
- [0102] 图80是根据本发明有利实施例用于下传数据的消息流图示；
- [0103] 图81是图解说明根据本发明有利实施例当文件被部分地传输时的消息流图示；
- [0104] 图82是图解说明根据本发明有利实施例的上传过程的消息流图示；
- [0105] 图83是图解说明根据本发明有利实施例的上传过程中的消息流的图示；
- [0106] 图84是根据本发明有利实施例上传数据的过程流程图；
- [0107] 图85是根据本发明有利实施例下传数据的过程流程图；
- [0108] 图86是图解说明根据本发明有利实施例的打包工具的图示；
- [0109] 图87是图解说明根据本发明有利实施例的打包工具的图示；
- [0110] 图88是图解说明根据本发明有利实施例的包处理的消息流图示；
- [0111] 图89是图解说明根据本发明有利实施例的打包工具的用户界面的一个实现的图示；
- [0112] 图90是图解说明根据本发明有利实施例检验包和将包解包的过程中的数据流图示；
- [0113] 图91是图解说明根据本发明有利实施例创建包的过程中的数据流的图示；

[0114] 图92是图解说明根据本发明有利实施例处理被接收的包的过程流程图。

具体实施方式

[0115] 现在参考附图,尤其是参考图1,给出了根据有利实施例的数据处理环境的示例性图示。应该意识到图1仅仅是示例性的并且没有声称或暗示对其中可实施不同实施例的环境的任何限制。如同在这里所使用的,术语示例性是指例子并且该例子不一定是理想的实施。可以对所描述的环境进行多种修改。

[0116] 现在转到图1,图1是根据有利实施例描述图解说明网络数据处理系统的图示,在该数据处理系统中可以实施软件部分管理环境。在该例子中,网络数据处理系统100是在其中信息可以在飞行器网络101和地面网络103之间传输的网络数据处理系统。该信息可包括,例如而不限于命令、软件飞行器部件、下行链路数据、错误日志、使用历史、飞行日期、状态信息和使用手册。地面网络103包括位于地面的网络和计算机。飞行器网络系统101是位于飞行器上的网络和计算机。

[0117] 在这些例子中,命令可以产生于位于库服务器计算机104上的库102。库服务器计算机104和其他数据处理系统,例如服务器计算机105和106都连接到内联网108。

[0118] 这些命令可以通过网络(诸如互联网112)分配到在代理服务器计算机110上的地面组件(OGC)109。内联网108和互联网112可以包括多个连接,例如电线、光纤光缆或无线通信链接。代理服务器计算机110可以位于机构例如机场114中。代理服务器,例如代理服务器计算机110可以位于其他的机场和其他的地点,例如维修地点。代理服务器计算机110为从库102接收的命令和部件提供临时部件存储器111。

[0119] 命令和软件飞行器部件也可以被发送到便携式计算机上的软件维护工具,例如维护笔记本计算机116上的软件维护工具115。代理服务器计算机110和维护笔记本计算机116共同被称为地面工具。地面工具可以是任何数据处理系统,该数据处理系统被合适的应用程序配置以传输信息,诸如命令、软件飞行器部件和下行链路数据。

[0120] 代理服务器计算机110可以通过不同类型的连接或链接连接到飞行器118。例如,无线单元120可以与飞行器118上的无线单元124建立无线连接122。无线单元124连接到飞行器118中的开放式数据网络126。维护笔记本计算机134具有软件维护工具136和地面组件(OGC)138并且可以建立与机舱无线访问单元142的通信链接140与飞行器118通信。该通信链接140是无线虚拟专用网络通路。机舱无线访问单元142在这些例子中连接到开放式数据网络126。开放式数据网络126为不同的通信链接(诸如无线链接122)提供接口。此外,连接到机场114的代理服务器计算机110的卫星单元128可以与卫星单元132建立卫星链接130,该卫星单元132同样也连接到开放式数据网络126。

[0121] 开放式数据网络126连接到飞行器数据处理系统144,其包含机载电子分配系统(OBEDS)146。存储装置148也位于飞行器数据处理系统144中。存储装置148提供存储信息例如飞行器部件的位置。飞行器数据处理系统144也包括文件传输系统(FTS)150、机载存储管理器(OSM)152、机载数据加载功能块(ODLF)154和签名者—打包者模块(SCM)156。在这些例子中,SCM模块156可以实现成编译到机载电子分配系统146中的Java[®]库。同样地,飞行器数据处理系统144可以采用成员信息系统/维护系统计算机的形式。

[0122] 文件传输系统150用于将文件从存储装置148向外场可更换单元传输。机载存储管

理器152管理存储在存储装置148中的信息。机载数据加载功能块154是用于将软件飞行器部件加载到外场可更换单元的软件组件。签名者一打包者模块156用于处理进入的包和将这些包中的内容存储在存储装置148中。此外,签名者一打包者模块156可以将下载数据打包,用于下载到代理服务器计算机110。

[0123] 在这些例子中,所有的命令处理都由位于飞行器数据处理系统144中的机载电子分配系统146启动。机载电子分配系统146监视空对地链接状态并且确定是否已经建立通信链接。如果链接可用,那么机载电子分配系统146通过此链接连接到地面数据处理系统。

[0124] 在其他有利实施例中,维护笔记本电脑158可以与隔离的数据网络166建立通信链接164。维护笔记本电脑158具有软件维护工具160和地面组件162。通信链接164可以是有线连接。外场可更换单元可以例如是中央服务器模块(CSM)168、电子飞行包(EFB)170和机舱服务系统(CSS)172。中央服务器模块168为飞行器118上的不同网络提供常规联网功能。这些服务包括:例如包路由、防火墙和无线接入。机舱服务系统172为飞行器上的控制系统提供应用,例如照明系统、舱门系统和地址系统。

[0125] 如果机载电子分配系统146建立与地面装置的连接,那么机载电子分配系统146请求为飞行器118排队或存储的一系列命令。在这些例子中,数据处理系统(诸如代理服务器计算机110、维护笔记本电脑134和/或维护笔记本电脑162)上的机载地面组件109、138或162与飞行器数据处理系统144上的机载电子分配系统136通信。这种类型的软件组件为地面工具提供应用程序接口以向飞行器118上传命令和软件飞行器部件以及下传数据或文件。

[0126] 对网络数据处理系统100中的特定组件和配置的图解说明并不意图暗示对不同实施例的实现方式的架构限制。例如,尽管在飞行器网络101中只显示了一个飞行器,但在飞行器网络101中也有可能存在多个飞行器。另一个例子,在地面网络103中的航空公司网络108可以通过不同于互联网112的其他类型网络连接到计算机,例如在机场(诸如机场114)的代理服务器计算机110。例如,广域网(WAN)可以被用于替代互联网112或者与互联网112结合。

[0127] 转到图2,根据有利实施例描述数据处理系统的图示。在这些例子中,数据处理系统200是可用于实现多个数据处理系统的数据处理系统的例子,该数据处理系统例如为图1中的库服务器计算机104、维护笔记本电脑116、代理服务器计算机110、维护笔记本电脑134、维护笔记本电脑158和飞行器数据处理系统144。

[0128] 在该说明性例子中,数据处理系统200包括通信组织202,该通信组织202提供处理器单元204、内存206、持久性存储器208、通信单元210、输入/输出(I/O)单元212和显示器214之间的通信。

[0129] 处理器单元204用于执行加载到内存206中的软件的指令。取决于特殊的实现,处理器单元204可以是一组一个或更多个处理器或者也可以是多处理器核。进一步地,处理器单元204可以使用一个或更多个异类处理器系统实现,在该系统中,主处理器与次处理器都在单个芯片上。作为另一个说明性实施例,处理器单元204可以是包含多个相同类型的处理器的对称多处理器系统。

[0130] 在这些例子中,内存206可以例如是随机存取存储器或任何其他合适的易失性或非易失性存储装置。持久性存储器208可以依据特殊的实现而采用不同的形式。例如,持久

性存储器208可以包含一个或更多个组件或装置。例如,持久性存储器208可以是硬盘驱动器、闪存、可擦写光盘、可擦写磁带或上述装置的某个组合。持久性存储器208所使用的介质也可以是可拆卸的。例如,可拆卸硬盘驱动器可以被用于持久性存储器208。

[0131] 在这些例子中,通信单元210提供与其他数据处理系统或装置的通信。在这些例子中,通信单元210是网络接口卡(网卡)。通信单元210可以通过使用物理和无线通信链接两者之一或同时使用物理和无线通信链接提供通信。输入/输出单元212允许与连接到数据处理系统200的其他装置输入和输出数据。例如,输入/输出单元212可以通过键盘和鼠标为用户输入提供连接。进一步地,输入/输出单元212可以向打印机发送输出。显示器214提供向用户显示信息的机构。

[0132] 用于操作系统和应用程序或程序的指令位于永久性存储器208。这些指令可以被加载到内存206中用于由处理器单元204执行。处理器单元204可以使用计算机实现的指令执行不同实施例的处理,指令可以位于内存中(诸如内存206)。这些指令被称为程序代码、计算机可用的程序代码或计算机可读的程序代码,该代码可以被处理器单元204中的处理器读取和执行。在不同实施例中的程序代码可以包含在不同的物理或有形的计算机可读介质上,例如内存206或永久性存储器208。

[0133] 程序代码216以功能形式位于计算机可读介质218上并且可以被加载到或传输到数据处理系统200以供处理器单元204执行。在这些例子中,程序代码216和计算机可读介质218形成计算机程序产品220。在一个例子中,计算机可读介质218可以具有有形的形式,例如光盘或磁盘,光盘或磁盘可以被插入或置于作为永久性存储器208的一部分的驱动器或其他装置中以用于传递到存储装置上,例如作为永久性存储器208的一部分的硬盘驱动器。在有形形式中,计算机可读介质218也可以采用永久性存储器208的形式,例如连接到数据处理系统200的硬驱动或闪存。计算机可读介质218的有形形式也称为计算机可记录存储介质。

[0134] 替代地,程序代码216可以通过与通信单元210的通信链接和/或通过输入/输出单元212的连接从计算机可读介质218被传输到数据处理系统200。通信链接和/或连接在说明性例子中可以是物理的(实体的)或无线的。计算机可读介质也可以采用非有形介质的形式,例如包含程序代码的通信链接或无线传输。

[0135] 为数据处理系统200所说明的不同组件并不意图提供对不同实施例的实现方式的架构上的限制。不同的说明性实施例可以实现在数据处理系统中,该数据处理系统包括除为数据处理系统200图解说明的那些组件以外的组件或替代为数据处理系统200图解说明的那些组件的组件。其他显示在图2中的组件可以与所显示的说明性例子不同。

[0136] 例如,可以使用总线系统实现通信组织202并且该总线系统由一个或更多总线组成,例如系统总线或输入/输出总线。当然,总线系统可以使用任何适当类型的架构实现,该架构提供在连接到总线系统的不同组件或装置之间的数据传递。此外,通信单元可以包括一个或更多个用于传输和接收数据的装置,例如调制解调器或网络适配器。此外,内存例如可以是内存206或诸如在存在于通信组织202中的接口或内存控制器集线器中的缓存。

[0137] 不同的有利实施例提供了计算机实现的方法、设备和计算机可用的程序产品,用于管理软件飞行器部件。

[0138] 现在参考图3,根据本发明有利实施例描述图解说明软件飞行器部件管理设备的

图示。在本例子中,软件飞行器部件管理设备300包括接收模块302、库304、代理服务器应用程序306、软件维护工具308和机载电子分配系统310。

[0139] 接收模块302能够从源接收软件飞行器部件并且向库304发送软件飞行器部件以进行存储。该源可以包括,例如,飞行器制造商、软件供应商、库供应商或航空公司。

[0140] 在这些例子中,库304位于数据处理系统上,例如图1中的库服务器计算机102。库304为软件飞行器部件提供存储系统。同样地,库304也可以用于管理软件飞行器部件。举例而非限制性的,对部件的管理可以包括:组织软件飞行器部件、删除软件飞行器部件和分配软件飞行器部件。安全控制处理以及版本化控制处理可用于管理软件飞行器部件。

[0141] 代理服务器应用程序306可以根据特定的实施而位于相同的数据处理系统或不同的数据处理系统上。代理服务器应用程序306与库304通信并且能够为不同的飞行器客户系统服务。

[0142] 软件维护工具308可以是位于便携式计算机上的软件维护工具,该便携式计算机提供从库304向机载电子分配系统310发送软件飞行器部件的替代途径。软件维护工具308根据特定的实施可以直接从库304或者通过代理服务器应用程序306接收软件飞行器部件。

[0143] 机载电子分配系统310是位于飞行器上的飞行器客户系统的例子。机载电子分配系统310是在飞行器上的数据处理系统上执行的软件客户系统。机载电子分配系统310可以通过代理服务器应用程序306从库304为飞行器接收软件飞行器部件。当机载电子分配系统310接收软件飞行器部件后,软件飞行器部件可以被安装在外场可更换单元中以供使用。

[0144] 除了使用软件飞行器部件管理设备300向飞行器分配软件飞行器部件以外,该设备也可以被用于接收飞行器产生的数据。该数据也被称为下行链路数据。例如,飞行记录器可以生成描述在飞行期间发生的不同事件的数据。该数据可以通过代理服务器应用程序306和/或软件维护工具308通过机载电子分配系统310被下传回库304,用于之后的使用和分析。该数据也可以包括关于软件飞行器部件、外场可更换单元或飞行器配置的配置数据。

[0145] 现在参考图4,根据有利实施例描述软件部件管理环境的框图。软件管理环境400是图3中的软件飞行器部件管理设备300的一个实施的例子。

[0146] 在本例子中,打包工具402在计算机404上运行。打包工具402是图3中接收模块302的一个实现的例子。库406位于服务器计算机410上。库406是图3中库304的一个实现的例子。代理服务器应用程序412在服务器计算机414上运行并且是图3中代理服务器应用程序306的一个实现的例子。软件维护工具416在便携式计算机418上运行。软件维护工具416是图3中软件维护工具308的例子。机载电子分配系统420运行在飞行器424上的飞行器计算机422上。机载电子分配系统420是图3中的机载电子分配系统310的一个实现的例子。

[0147] 在这些例子中,打包工具402可以接收和处理软件部件,例如包428中的软件飞行器部件426。包428是软件飞行器部件426的打包系统并且不是实体物体。在这些例子中,包428是包含软件飞行器部件426的文件。包428例如可以是使用zip文件格式的zip文件。

[0148] zip文件格式是数据压缩和存档格式,在该zip文件格式中,zip文件可以包括一个或多个被压缩的文件。文件打包系统的其他例子例如可以是Java[®](JAR)档案文件。根据特定的实施,这些文件也可以被加密或数字签名。当然,可以使用为软件飞行器部件426提供包装器的任何类型的机制。在这些例子中,包装器是安全包装,其被设计为满足为软件飞行器部件426设定或要求的不同安全要求。

[0149] 在这些例子中,软件飞行器部件426可以是用于飞行器中的数据处理系统上的软件应用程序。数据处理系统可以是位于飞行器424中的外场可更换单元430内的数据处理系统。应用程序可以包括一组文件。该组文件例如可以包括:一个或多个程序、数据文件、库、配置文件或其他信息或代码。这里所使用的“一组”指一项或更多项。作为例子,在这些例子中,一组软件飞行器部件是一个或多个软件飞行器部件并且一组命令是一个或多个命令。

[0150] 打包工具402接收包428并且处理包428以将软件飞行器部件426存储到库406中的软件飞行器部件组432中。该处理例如可以包括为包428和软件飞行器部件426验证签名。该验证可以被执行以确保在包428或软件飞行器部件426中没有损坏和错误。存储在库406内的不同部件可以通过代理服务器应用程序分配给飞行器(诸如飞行器424)。

[0151] 库406在软件飞行器部件管理环境400中提供组件以对软件飞行器部件组432进行不同的管理操作。举例而非限制的,这些管理操作可以包括将软件飞行器部件分配到飞行器、组织软件飞行器部件、删除软件飞行器部件、从存在软件飞行器部件的飞行器接收数据并且接收新的软件飞行器部件。

[0152] 代理服务器应用程序412可以从软件飞行器部件组432获取一组软件飞行器部件并且将这些部件发送到机载电子分配系统420。代理服务器应用程序412通过通信链接与机载电子分配系统420通信。该通信链接可以采用不同的形式。例如,可以使用无线通信链接。以这种方式,当飞行器在地面上或者甚至在飞行时飞行器部件和数据可以被交换。在其他例子中,服务器计算机414可以通过网络中的有线链接连接到服务器计算机422。

[0153] 机载电子分配系统420处理该组飞行器部件并且将这些部件作为软件飞行器部件434存储在飞行器计算机422上的存储装置436中。如果需要的话,来自软件飞行器部件434的软件飞行器部件可以被安装在外场可更换单元430中。被发送到飞行器的数据,诸如软件飞行器部件、使用手册、文档和命令被称为上行链路数据。

[0154] 此外,数据可以在另一个方向通过代理服务器应用程序412从飞行器424流回库406。这种类型的数据被称为下行链路数据。在这些例子中,外场可更换单元430可以生成暂时存储在存储装置436中的下行链路数据438。机载电子分配系统420可以将下行链路数据438发送到代理服务器应用程序412。随后,代理服务器应用程序412将下行链路数据438发送到库406用于存储。该数据接着可以被处理和分析。该数据也可以包括,例如飞行器上软件的状态。该状态信息可以被用于将操作器发送到飞行器以启动将外场可更换单元加载和安装到飞行器上。

[0155] 此外,在便携式计算机418上的软件维护工具416提供用于传输软件飞行器部件和下行链路数据的替代途径。便携式计算机418例如可以是笔记本计算机。根据特殊的实现,便携式计算机418可以通过代理服务器应用程序412从软件飞行器部件组432获取软件飞行器部件或者直接从库406获取软件飞行器部件。因此,便携式计算机418可以被传输到飞行器424并且与飞行器计算机422上的机载电子分配系统420建立通信链接以向机载电子分配系统420发送软件飞行器部件。

[0156] 当在服务器计算机414和飞行器424上的飞行器计算机422之间无法建立网络连接或通信链接时,这种类型的软件飞行器部件的分配特别地有用。这种情况的发生取决于机场或维修机构中可使用的装备的类型。此外,在某些情况下,提供通信链接的网络或通信系

统可能会暂时不可以使用或需要维修。以这种方式,软件维护工具416将软件飞行器部件传输到机载电子分配系统420。此外,软件维护工具416在与机载电子分配系统420通信时也可以接收下行链路数据438。

[0157] 这样,不同的有利实施例提供计算机实现的方法、设备和计算机可用的程序代码,用于管理软件飞行器部件。此外,不同的有利实施例也将数据从飞行器传输到机构或位置以用于之后的分析或审查。

[0158] 现在转到图5,根据有利实施例描述图解说明软件部件管理环境的操作模式的表格。在本例子中,表格500图解说明不同的操作模式中的一些操作模式,其发生在图4中的软件部件管理环境400中。在这些例子中,不同的操作模式包括接收和存储部件模式502,分配命令模式504、分配部件模式506和接收下行链路数据模式508。图解说明在表格500中的这些不同的操作模式是可能发生在图4中的软件部件管理环境400中的一个或更多个组件中的模式。

[0159] 在接收和存储部件模式502中,软件飞行器部件可以被接收和存储在图4中的库406中。分配命令模式504被用于向飞行器发送命令。这些命令可以是例如上行链路数据。该数据可以包括,例如,向飞行器发送软件飞行器部件。上行链路数据也可以包括向飞行器发送文档或其他信息的命令。分配部件模式506是其中软件飞行器部件被实际发送到飞行器的操作模式。接收下行链路数据模式508是其中将数据从飞行器中的不同组件发送到软件部件管理环境中的库的操作模式。

[0160] 命令协议处理细节

[0161] 现在参考图6,图6根据有利实施例描述图解说明命令类型的图示。在该例子中,命令类型600包括上行链路命令602、下行链路命令604和删除命令606。上行链路命令602用于从库向飞行器发送信息。该信息可以包括,例如,软件飞行器部件、配置信息和其他数据。下行链路命令604用于启动从飞行器向库传输数据。该信息可以包括,例如,关于软件飞行器部件的上传的状态信息和配置飞行器上的外场可更换单元的报告。删除命令606用于删除飞行器上的信息。例如,删除命令606可以用于删除飞行器上选中的软件飞行器部件。在这些例子中,在包中将不同的命令发送到飞行器。

[0162] 现在转向图7,图7示出根据有利实施例的命令格式。在该例子中,命令700采用可扩展标记语言(XML)数据结构。在该例子中,命令700是上行链路命令。

[0163] 命令700中的消息标识符元素702提供对命令的唯一标识符。类型元素704指出命令的类型。在该例子中,命令的类型被标识为上行链路命令。系统元素706标识命令的目标系统。应用程序标识符元素708标识接收命令的目标系统上的应用程序。

[0164] 链接标签元素710标识用于将命令从库传输到飞行器的网络链接的类型。例如,链接可以是有线链接或无线链接。服务器地址元素712标识被标识装置的地址。数据类型元素714提供从属于命令的信息类型的标识。资源类型元素716标识从属于命令的特定文件。

[0165] 现在转向图8,图8根据有利实施例描述图解说明上行链路命令的处理的消息流图示。在该例子中,上行链路命令的处理涉及地面系统800、机载电子分配系统(OBEDS) 802、文件传输系统(FTS) 804和外场可更换单元(LRU) 806。在这些例子中,地面系统800例如是在计算机上的代理服务器应用程序或位于笔记本计算机上的软件维护工具。

[0166] 该过程始于机载电子分配系统802建立与地面系统800的连接(消息M1)。响应该连

接,地面系统800激活下一个命令。在该例子中,下一个命令是上行链路命令。地面系统800发送上行链路命令给机载电子分配系统802(消息M2)。机载电子分配系统802检查上行链路命令的签名。

[0167] 之后,机载电子分配系统802向地面系统800发送请求,请求资源(消息M3)。地面系统800使对应于该请求的包可用于传输。在消息M3中的请求是从消息M2中的接收的从上行链路命令中被识别的。

[0168] 机载电子分配系统802从地面系统800上传包(消息M4)。在接收到包之后,机载电子分配系统802验证包上的签名。该验证包括验证包上的签名,同时也验证软件飞行器部件的签名。

[0169] 此后,机载电子分配系统802向文件传输系统804发送传输请求(消息M5)。作为响应,传输系统804将软件飞行器部件传输到外场可更换单元806(消息M6)。

[0170] 接着状态从文件传输系统804被传输到机载电子分配系统802(消息M7)。

[0171] 现在转向图9,图9根据有利实施例描述图解说明下行链路命令的处理的消息传送图示。在该例子中,存在和图8中相同的组件用于处理下行链路命令。在该例子中,机载电子分配系统802连接到地面系统800(消息N1)。地面系统800使下一个命令可用于处理。

[0172] 下行链路命令被发送到机载电子分配系统802(消息N2)。机载电子分配系统802向文件传输系统804发送请求,以向外场可更换单元806发送下行链路命令(消息N3)。接着,文件传输系统804向外场可更换单元806发送下行链路命令(消息N4)。外场可更换单元806处理该命令并且接着向文件传输系统804发送下行链路数据(消息N5)。文件传输系统804向机载电子分配系统802发送请求,以将下行链路数据发送到地面系统800(消息N6)。作为响应,机载电子分配系统802将下行链路数据打包并且签名。此外,机载电子分配系统802也在包中加入元数据。此后,机载电子分配系统802向地面系统800发送包(消息N7)。

[0173] 现在转向图10,图10根据有利实施例示出图解说明删除命令的处理的消息流图示。和在图8和图9中所描述的组件相同的组件也用于处理删除命令。该过程始于将机载电子分配系统802连接到地面系统800(消息O1)。作为响应,地面系统800使下一个命令可由机载电子分配系统802使用。机载电子分配系统802接收删除命令(消息O2)。

[0174] 此后,机载电子分配系统802检验命令的签名。如果签名是有效的,则机载电子分配系统802向文件传输系统804发送请求,以向外场可更换单元806发送删除命令(消息O3)。在这些例子中,只有在命令是发送自代理服务器的时候才检验命令上的签名。对于下行链路命令也是这样。此后,文件传输系统804将删除命令发送到外场可更换单元806(消息O4)。

[0175] 响应于接收到删除命令,外场可更换单元806删除由删除命令标识的资源。

[0176] 文件传输系统804向机载电子分配系统802发送请求,以向地面系统800发送状态(消息O5)。该状态指出资源是否已经由外场可更换单元806成功删除。作为响应,机载电子分配系统802将该系统发送到地面系统800(消息O6)。响应收到该请求,机载电子分配系统802将状态打包并且签名。此后,包被发送到地面系统800(消息O7)。

[0177] 现在转向图11,图11根据有利实施例示出用于分配软件飞行器部件的过程的高级流程图。

[0178] 在图11中图解说明的过程是可在图4中的软件部件管理环境400中找到的将软件飞行器部件安装到飞行器上的过程的例子。

[0179] 该流程开始于将软件飞行器部件存储到库中(操作1100)。在这些例子中,库是软件飞行器管理组件,例如图4中的库406。该流程接着识别飞行器以接收软件飞行器部件从而形成被识别的飞行器(操作1102)。在操作1102中,库的操作员可以选择用于分配给一组飞行器的软件飞行器部件。在其他实施例中,可以通过通信或从其他系统接收的文件来预选软件飞行器部件的目标飞行器。

[0180] 此后,以上行链路命令的形式向代理服务器应用程序发送软件飞行器部件(操作1104)。代理服务器应用程序将上行链路命令和软件飞行器部件发送到被识别的飞行器上的机载电子分配系统(操作1106),此后终止流程。

[0181] 现在转向图12,图12根据有利实施例示出接收和存储软件飞行器部件的流程图。图解说明在图12中的过程可以在图4中的软件部件管理环境400中实现。该过程是图5中接收和存储部件模式中执行的步骤的例子。

[0182] 该流程开始于打包工具接收包(操作1200)。该工具可以是,例如在图4中的打包工具402。在该例子中,包包括可从源点(诸如飞行器制造商或飞行器中的外场可更换单元)请求得到的软件飞行器部件。响应软件飞行器部件可用的通知,该软件飞行器部件可以被接收并且通过某传输机制被传递。包可以在物理介质或有形介质诸如光盘、闪存或数字通用光盘上接收。在其他实施例中,包可以通过传输介质,例如网络上的通信链接接收。

[0183] 打包工具验证并且将包拆包(操作1202)。在该操作中,如果签名是无效的或者摘要与打包工具所计算的摘要不匹配,则生成通知。如果没有探测到问题,则包被拆包到不同的位置以进行另外的处理。接着,打包工具验证软件飞行器部件的签名(操作1204)。如果签名是无效的或者软件飞行器部件的摘要与所计算的摘要不匹配,则生成通知。如果没有探测到问题,则在包的内容被验证或校验后该部件准备好被签名。

[0184] 打包工具接着检验包内容(操作1206)。在本操作中,包的内容可以向检验者显示,以供检验者校验该内容。在其他实施例中,本操作可以自动进行,用来比较该内容与标识包期望内容的文件或配置信息。

[0185] 一旦内容已经被校验,打包工具以航空公司签名为软件飞行器部件签名(操作1208)。根据实现,也可以使用其他实体的签名。例如,该签名可以是客户的签名或管理库的其他组织的签名。如果在为软件飞行器部件签名过程中没有发生错误,则该部件准备好被存储。

[0186] 此后,打包工具将软件飞行器部件置于库中(操作1210),此后该流程结束。该操作包括将软件飞行器部件从其在文件系统上的当前位置转移到包含不同软件飞行器部件的库的存储区域。在这些例子中,库可以例如是图4中的库406。

[0187] 参考图13,图13根据有利实施例示出通过代理服务器分配命令的过程流程图。图13中图解说明的过程可以在图4中的软件部件管理环境400中实现。具体地,该过程是在图5中的分配命令模式504期间执行的过程的例子。

[0188] 该流程开始于代理服务器应用程序接收和处理排队的命令(操作1300)。在这些例子中,排队的命令以包发送,称为命令包。该包被处理并且发送到库中合适的飞行器命令队列中。代理服务器应用程可以自库的队列中访问和检索排队的命令。如果这些命令包括上行链路命令,那么打包的软件飞行器部件也被置于代理服务器的本地清单中。在这些例子中,命令被置于包中以分配给代理服务器。

[0189] 此后,代理服务器应用程序连接到飞行器上的机载电子分配系统(操作1302)。代理服务器应用程序可以同时连接到多个飞行器。在这些例子中,飞行器通过无线连接或通信链接连接到代理服务器应用程序。一旦建立通信链接,信息可以在代理服务器应用程序和机载电子分配系统之间传输。该信息可以包括,例如,命令、数据、软件飞行器部件、配置文件、使用手册和状态信息。

[0190] 代理服务器应用程序接着自动将针对飞行器的打包命令传输到机载电子分配系统(操作1304)。在这些例子中,指定给飞行器的打包命令可供机载电子分配系统用于检索。

[0191] 机载电子分配系统读取命令并且执行命令(操作1306)。在这些例子中,机载电子分配系统轮询代理服务器应用程序上的命令队列并且每次为一个命令检索飞行器的每个命令。机载电子分配系统接着检验打包的命令(操作1308)。如果包经过检验,那么命令接着传递到指定的系统和应用程序。此后,机载电子分配系统返回命令传输的状态(操作1310),并且此后该过程结束。

[0192] 接着转向图14,图14根据有利实施例示出通过代理服务器应用程序接收和分配下行链路数据的过程流程图。在图14中图解说明的过程可以在图4中的软件部件管理环境400中实现。具体地,图14中图解说明的过程是发生在图5的接收下行链路数据模式508期间的操作的例子。

[0193] 该流程始于代理服务器应用程序连接到飞行器上的机载电子分配系统(操作1400)。此后,代理服务器应用程序接收数据下行链路(操作1402)。在这些例子中,机载电子分配系统为在包含下行链路数据的队列中的每一项生成下行链路。

[0194] 此后,代理服务器应用程序将下行链路数据置于本地清单中(操作1404)。该下行链路数据被存储,以基于某个事件传输回到库。在这些例子中,事件可以是周期性事件,例如计时器终止。在其他例子中,事件可以是非周期性的事件,例如用户生成的请求。然后,代理服务器应用程序向库发送下行链路数据(操作1406),此后该流程结束。在操作1406中,下行链路数据被置于目录中,供以后使用或分析。

[0195] 现在转向图15,图15根据有利实施例示出使用软件维护工具分配软件飞行器部件的过程流程图。图15中图解说明的过程可以在图4中的软件部件管理环境400中实现。在该过程中的不同操作是发生在图5中的分配部件模式506期间的操作的例子。

[0196] 该流程开始于软件维护工具连接到网络(操作1500)。该网络是库存在于其中的网络。在该例子中,部件并不存在和位于软件维护工具上。下一步,软件维护工具检索一组上行链路命令并且与软件飞行器部件一起打包(操作1502)。

[0197] 此后,软件维护工具从网络断开连接(操作1504)。软件维护工具接着被移走并且连接到飞行器上的机载电子分配系统(操作1506)。在这些例子中,该连接需要操作人员来启动(发起)该连接。软件维护工具自动地将该组上行链路命令传输到机载电子分配系统(操作1508)。在这些例子中,以一次一条命令的形式将命令发送到机载电子分配系统。在操作1508中每次发送一条命令,在发送下一条命令前检查机载电子分配系统是否已经完成上传命令或其他信息。

[0198] 机载电子分配系统读取该组命令并且接收包含软件飞行器部件的包组(操作1510)。在这些例子中,机载电子分配系统从软件维护工具以一次一条的形式检索该命令组中的每一条命令。机载电子分配系统检验包组(操作1512)。如果包被检验,则继续传递软件

飞行器部件以存储并且在飞行器中分配。

[0199] 接着,机载电子分配系统将传输的状态返回给软件维护工具(操作1514)。软件维护工具接着返回传输状态(操作1516),此后该过程结束。在该例子中,软件维护工具将状态返回到软件飞行器部件源,例如库或代理服务器应用程序。

[0200] 在这些例子中,可以手工添加上行链路命令而不是从库自动地接收。例如,软件维护工具的操作者选择传输到飞行器的软件飞行器部件。该选择会导致软件维护工具产生传输软件飞行器部件的适当命令。该过程还接收软件飞行器部件组的包。

[0201] 然而,在这种类型的实现中,该流程从操作1508继续进行以接收选择的软件飞行器部件(操作1518)。在这些例子中,该选择是基于用户输入。此后,软件维护工具向机载电子分配系统发出上行链路命令(操作1520)。该命令可以被置于命令队列中供机载电子分配系统检索。

[0202] 现在转向图16,图16根据有利实施例示出使用软件维护工具接收数据的过程流程图。图16中图解说明的过程可以使用图4中的软件部件管理环境400实现。图16中图解说明的操作是在图5中的接收下行链路数据模式508期间发生的操作的例子。

[0203] 该流程开始于软件维护工具连接到机载电子分配系统(操作1600)。软件维护工具从机载电子分配系统接收数据下行链路(操作1602)。机载电子分配系统为其下行链路数据队列中的每一项启动下行链路。

[0204] 此后,软件维护工具将数据置于本地清单中(操作1604)。在该操作中,软件维护工具接受该下行链路并且将数据置于该链路的清单中。其他与该下行链路数据有关的数据可以显示在用户界面中。该用户界面允许用户使用不同的参数将下行链路数据排序为经过过滤的下行链路数据。这些参数可包括例如而限于飞行器标识、系统标识、应用程序标识或数据类型。

[0205] 下一步,软件维护工具从机载电子分配系统断开连接(操作1606)。软件维护工具从飞行器转移到另一个位置以传输下行链路数据。软件维护工具连接到网络(操作1608)。软件维护工具接着向库发送下行链路数据(操作1610),此后该过程结束。

[0206] 库

[0207] 图17-33描述了在软件部件管理环境下的库。具体地,这些图图解说明了在图4中的软件部件管理环境400中实现库406的一个例子。

[0208] 参考图17,图17根据有利实施例示出库的功能框图。库1700是图4中的库406的更详细例子。库1700包括用户界面系统1702和后端1704。后端1704包括接口1706、存储器1708和管理组件1710。

[0209] 接口1706包括消息收发服务1712、超文本传输协议(HTTP)服务1714和Web网络服务1716。存储器1708包括文件系统1718和数据库1720。在这些例子中,管理组件1710包括部件室1722、命令调度器1724、命令队列管理器1726、系统配置器1728、输入文件集合器1730、报告管理器1732、事件记录器1734和安全管理器1736。

[0210] 用户界面系统1702为操作员提供对后端1704的访问,以进行不同的任务和操作。用户界面系统1702可以是图形用户界面。更具体地,用户界面系统1702可以是基于网络的应用程序,其允许用户从远程位置访问库1700。接口1706包括多个用于将信息输入到库1700和从库1700输出信息不同界面的。在接口1706中,消息收发服务1712允许管理组件

1710内的各种组件和其他应用程序通信。在这些例子中,报告管理器1732使用消息收发服务1712分配报告以响应请求。消息收发服务1712可以使用各种类型的消息收发服务实现。例如,消息收发服务1712可以使用Java[®]消息收发服务实现,Java[®]消息收发服务是Java[®]2企业版套装中的一部分。该产品由Sun微系统(Sun Microsystems)有限公司提供。

[0211] Web网络服务1716可以使用任何Web网络服务系统实现。Web网络服务1716被设计为提供库1700和网络上的其他装置之间的交互。通过使用可通过网络(诸如因特网)访问的应用程序接口来实现接口1706中的Web网络服务1716。Web网络服务1716可以使用各种协议来实现,例如简单对象访问协议(SOAP)或网络服务描述语言(WSDL)。超文本传输协议(HTTP)服务1714可以被实现以提供请求和响应系统来管理客户的响应。这些请求一般称为HTTP请求。

[0212] 在这些例子中,超文本传输协议服务1714可以用于发送和接收信息,例如文件。这些文件例如可以是:包含软件飞行器部件的文件、命令、下行链路数据和其他合适的信息。在这些例子中,超文本传输协议服务1714被部件室1722、命令调度器1724、输入文件集合器1730、报告管理器1732和事件记录器1734使用。

[0213] 如所描述的,来自于管理组件1710中组件的不同组件可以以不同的理由访问存储器1708。存储器1708包含不同的存储系统,其用于在库1700的后端1704中存储信息,例如文件系统1718和数据库1720。存储器1708是存储信息的功能组件并且可以位于一个或多个存储装置,例如硬盘驱动器或随机存取存储器上。存储器1708例如可以位于单个存储装置,诸如硬盘驱动器上。

[0214] 在其他实施例中,存储器1708可以位于多个存储装置上,该多个存储装置可以位于相同的物理地址或不同的物理地址。在存储器1708中,文件系统1718提供用于存储数据的结构或体系结构。该数据例如可以包括软件飞行器部件、文档、下行链路数据或其他文件。在这些例子中,数据库1720可以包含例如与位于文件系统1718中的文件相关的元数据和命令。此外,数据库1720可以包括用于执行其他功能的其他命令,该功能例如为删除飞行器上下载下行链路数据的文件。

[0215] 部件室1722提供管理存储和将软件飞行器部件分配到不同飞行器上的过程。具体地,部件室1722提供对部件的安全分配。这些过程可以接收新的软件飞行器部件并且将软件飞行器部件封包或打包,以分配到飞行器。

[0216] 命令队列管理器1726是管理飞行器命令的组件。命令队列管理器1726可以允许用户或操作员通过用户界面系统1702检查、记录和改变数据库1720中的命令的状态。命令的检查可以允许用户看到不同的命令或基于不同的标准来过滤命令。

[0217] 命令调度器1724是管理创建和调度命令的组件。该组件允许用户或操作员通过用户界面系统1702创建上行链路并删除上下行链路命令。命令调度器1724也提供在创建这些不同类型的命令时的输入参数确认。当外部装置以不同的标准请求时,该组件提供分组、打包和调度命令的机制。

[0218] 在这些例子中,系统配置器1728管理数据的配置以支持由命令调度器1724进行的操作。系统配置器1728允许用户定义、选择或输入信息以定义可连接到库1700的外部装置。此外,该组件允许为软件飞行器部件定义飞行器模型、特定飞行器和目的地系统。在这些例

子中,这些目的地系统可以包括位于飞行器中的外场可更换单元。

[0219] 输入文件集合器1730执行并发引入由外部装置发送到库1700的大量文件。报告管理器1732允许操作员定义由库1700生成的报告。这些报告可以是包括来自事件日志的信息的报告,该事件日志可以从和软件部件管理环境的操作相关的不同源集合。例如,报告管理器1732可以允许用户定义报告,该报告标识成功地将特定类型的软件飞行器部件上传到软件部件管理环境下管理的飞行器的特定模型。

[0220] 事件记录器1734记录关于库1700的操作的事件。此外,事件记录器1734可以从连接到库1700的不同装置集合日志。这些事件可以包括,例如而不限于从外部源接收的软件飞行器部件、软件飞行器部件到飞行器的成功传输、为上传数据生成的命令、为下载数据生成的命令以及产生的用于删除软件飞行器部件的命令。

[0221] 下一步,安全管理器1736提供一种机制,以管理操作员使用用户界面系统1702对库1700的访问。可以通过使用可为特定用户配置的任务和责任实现安全管理器1736。这种类型的访问可以提供访问库1700中的不同特征或功能的用户特权。进一步地,安全管理器1736也可以提供外部装置和库1700之间的安全通信。例如,安全管理器1736可以确保通过接口1706的通信通过诸如加密或虚拟专用网络的机制发生。

[0222] 在操作中,库1700可以从外部程序,例如在图4中的打包工具402接收软件飞行器部件。在这种类型的操作中,外部程序连接例如超文本传输协议服务1714的服务和接口1706。安全管理器1736执行对连接的鉴定并且确定软件飞行器部件是否可以被引入。如果连接被允许,那么超文本传输协议服务1714可以接着向部件室1722发送请求以处理输入过程。在该过程中,部件室1722将关于软件飞行器部件组的元数据写入到数据库1720中并且将文件系统1718中实际的软件飞行器部件组存储到某选定的文件目录下。

[0223] 在管理库1700中的部件时,通过部件室1722从文件系统1718中检索出软件飞行器部件并且将该软件飞行器部件发送到用户用于检查或审查。此外,软件飞行器部件可以被存档到文件系统1718中。这种类型的存档将部件保存在某指定的目录或其他存储装置中。此外,部件室1722也可以扫描文件系统1718中部件的证书,以识别将部件签名的任何证书是否已经过期。过期的通知通过用户界面系统1702预先产生。此外,证书过期也会导致部件室1722使任何包含该部件的命令失效。

[0224] 库1700也可以被用于创建命令和向外部装置分配命令。这些命令可以由操作员通过用户界面系统1702创建。用户界面系统1702允许用户为命令输入信息。一旦生成命令,则将这些命令存储在数据库1720中。

[0225] 当外部装置需要这些命令时,将这些命令打包并且通过接口1706发送。具体地,Web网络服务1716可以被用于将这些命令发送到外部组件,诸如代理服务器应用程序或软件维护工具。在这些例子中,如果软件飞行器部件是由命令指定的,那么这些部件可以以分开传输的方式通过超文本传输协议服务1714发送。根据特殊的实现方式,这些软件飞行器部件可以在被请求时发送或作为传输的一部分被发送。

[0226] 此外,报告是可存储在文件系统1718中的其他数据的例子。这些报告可以是,例如,电子表格、部件列表和实况报告。

[0227] 从下行链路数据获取的信息,诸如文件和装置日志可以被存储在文件系统1718中。这些文件可以使用输入文件集合器1730被集合。除了将文件保存在文件系统1718中以

外,该组件可以接受文件并且在数据库1720中创建元数据项。

[0228] 为库1700图解说明的不同组件作为用于不同功能的通信的一个例子给出。这些不同组件的描述和组织并不意味着暗示对组件实现方式的架构限制。例如,库1700中的不同组件可以以与所显示方式不同的方式再分组或组合。此外,在其他的实现中,某些功能可以被省略而另外的功能可以被添加。进一步地,某些功能可以被组合并且实现为库1700中的单个模块或应用程序。作为另一个例子,接口1706可以使用除了图解说明的接口以外的其他接口或替代图解说明的接口的其他接口来实现。

[0229] 现在转向图18,图18根据有利实施例示出图解说明文件系统目录布局的图示。文件系统目录布局1800是在图17的文件系统1718中使用的布局或方案的例子。在本例子中,文件系统目录布局1800定义用于在图17中的文件系统1718中定位文件的信息。文件类型包括,例如下行链路1802、日志1804、部件1806、替换部件签名列表 (APSL) 1808、电子表格1810和档案1812。

[0230] 在文件系统目录布局1800中依据不同类型的信息标识这些类型的文件中的每一种类型。例如,下行链路1082包括日期1814、装置1816、尾数1818、唯一标识符 (UID) 1820和下行链路文件名1822。日期1814标识下行链路文件的创建日期。装置1816标识将下行链路数据从飞行器传输到库的装置。该装置可以是,例如代理服务器应用程序或软件维护工具。尾数1818标识在其上安置有下行链路数据的特定飞行器。唯一标识符1820唯一地标识文件系统中的文件。下行链路文件名1822是下行链路文件的名称。

[0231] 下一步,日志1804包括装置1824、唯一标识符 (UID) 1826和事件日志文件名1828。部件1806用于软件飞行器部件并且包括唯一标识符 (UID) 1830、包1832、包文件名1834和打包的部件文件名1836。包1832标识具有软件飞行器部件的包所位于的目录。包文件名1834是包文件的名称。打包的部件文件名1836是包含软件飞行器部件的文件的名称。

[0232] 互换部件签名列表 (APSL) 1808包括唯一的名称1838并且电子表格1810包括唯一的名称1840。档案1812包括软件飞行器部件 (SAP) 1842、唯一标识符 (UID) 1844和包文件名1846。

[0233] 给出的文件系统目录布局1800作为在图17中的文件系统1718的一个实现的例子。在其他有利实施例中可以使用其他适用于特殊实现的文件系统布局或方案。

[0234] 现在参考图19,图19根据有利实施例示出图解说明队列中命令的组织的图示。在本例子中,队列1900、1902和1904是位于图17中的数据库1720中的队列例子。

[0235] 队列1900包括命令1906;队列1902包括命令1908;并且队列1904包括命令1910。在这些例子中,命令1906、1908和1910是去往特定飞行器的命令。命令可以是,例如上行链路命令、下行链路命令或删除命令。上行链路命令是从图17中的库1700向飞行器发送信息的命令而下行链路命令是从飞行器向图17中的库1700发送信息的命令。

[0236] 删除命令是用于删除飞行器上的信息的命令。该信息例如可以是,软件飞行器部件、配置文件或使用手册。在这些例子中,这些队列中的每一个都与一个特定的地面工具或装置相关。在这些例子中,队列1900、1902和1904与不同的装置关联或指派给不同的装置,所述装置将命令分配给飞行器。

[0237] 例如,队列1900可以与第一代理服务器应用程序关联,队列1902与第二代理服务器应用程序关联并且队列1904与软件维护工具关联。当不同的装置与图17中的库1700联系

时,命令被分配到哪个装置取决于命令是否存在于与所述装置关联的队列中。

[0238] 现在转向图20,图20根据有利实施例示出软件飞行器部件的框图。在该例子中,软件飞行器部件2000存储在包2002中。包2002存储在图17的文件系统1718中。

[0239] 在这些例子中,包2002是文件。包2002可以为,例如而不限于zip文件格式。包2002在一些实施例中也可以包含多于一个软件飞行器部件。软件飞行器部件2000可以包括一组为特定部件提供功能性的文件。这些文件例如可以包括,可执行文件、数据文件、配置文件和库文件。

[0240] 在所描述的实施例中,包2002和软件飞行器部件2000被签名。换句话说,软件飞行器部件2000被一个数字签名签名而包2002被另一个数字签名签名。根据特定的实施例,这些数字签名可以是相同的或不同的。当然,在其他的实现中,软件飞行器部件2000可以不存储在包2002中。

[0241] 现在参考图21至23,图21至23是根据有利实施例描述的命令数据结构的例子。图解说明在这些图中的不同命令数据结构是从存储在队列,例如图19中的队列1900、1902和1904中的命令创建的临时数据结构例子。

[0242] 现在转向图21,图21根据本发明有利实施例描述删除命令的命令数据结构。在这个例子中,删除命令数据结构2100包括参数2102、2104、2106、2108、2110和2112。

[0243] 参数2102标识要被删除的一组文件名。参数2104标识要被删除的一组部件标识符。参数2106是一组飞机标识符,其标识接收删除命令的特定飞行器。该列飞机标识符可以是例如尾数。参数2108标识要发送命令的一组装置。这些装置例如可以是地面工具,诸如代理服务器应用程序或软件维护工具。

[0244] 参数2110标识接收命令的目的地系统。在这些例子中,目的地系统是要接收命令的特定外场可更换单元。参数2112标识请求命令的用户。

[0245] 在图22中根据有利实施例描述图解说明上行链路命令的命令数据结构的图示。在该例子中,上行链路命令数据结构2200包括参数2202、2204、2206、2208和2210。参数2202标识要被上传或发送的软件飞行器部件。参数2204标识接收命令的一组飞机。这些参数包括飞行器标识符。参数2206是针对处理命令的装置的一组装置标识符。参数2208是标识接收命令的目的地系统的一组参数。参数2210标识请求命令的用户。

[0246] 下一步转向图23,图23根据有利实施例描述图解说明下行链路命令的数据结构的图示。在该例子中,下行链路命令数据结构2300包括参数2302、2304、2306、2308和2310。

[0247] 参数2302标识正在被下传的数据的类型。参数2304标识接收下传数据的命令的一组飞行器。参数2306针对一组将命令发送到飞行器组的装置。参数2308标识接收命令的飞行器组上的一组外场可更换单元。参数2310标识请求命令的用户。

[0248] 这些命令数据结构是命令的缩写形式,这些命令允许诸如代理服务器应用程序或软件维护工具的装置开始处理由命令数据结构引用的命令。在这些例子中,这些命令数据结构可以降低经过不同通信链接的通信总量。装置在接收到这些命令数据结构后可以请求实际的命令。在这些例子中,这些命令数据结构在被发送到地面工具后被删除。

[0249] 现在参考图24,图24根据有利实施例描述调度命令的用户界面的图示。窗口2400是用户界面的例子,该用户界面通过图17中的命令调度器1724的用户界面系统1702呈现。

[0250] 在该例子中,用户可以在创建命令,诸如上行链路命令和下行链路命令之间做选

择。可以通过控件2402和2404做出该选择。控件2402可以用来生成上行链路命令而控件2404可以用来生成下行链路命令。控件2406可用来生成删除命令。

[0251] 在这个描述的例子中,控件2402已经被选择,导致区域2408显示在窗口2400中。区域2408使用户能够输入信息以创建一个上行链路命令。例如,用户可以从列表2410中选择飞机尾数。这些飞机尾数对于特定的飞行器是唯一的。

[0252] 用户也可以从列表2412中选择代理服务器应用程序形式的装置以分配命令。同样地,也可以从列表2414中选择软件维护工具形式的装置。飞行器上的目的地系统可以通过栏2416选择。在这些例子中,目的地系统是特殊的外场可更换单元。栏2418允许输入部件号码。所输入的该部件号码提供了关于显示在栏2420、2422、2424、2426和2428中的部件的其他信息。根据特殊的实现,所显示的关于部件的特定信息可以变化。

[0253] 栏2416具有针对不同命令类型的不同的可选择值并且规则由栏1724来执行,栏1724表示在图17的控件1702和控件1704中。

[0254] 图25至26是根据有利实施例的图形用户界面的图示。这些图形用户界面是通过图17中的用户界面系统1702呈现的界面的例子。呈现这些被描述的图形用户界面是为了图解说明一个特殊实现而并不意图限制由图17中的用户界面系统1702设计或呈现图形用户界面的方式。

[0255] 转向图25,图25根据有利实施例示出图解说明浏览命令的用户界面的图示。在该例子中,窗口2500是图形用户界面的例子,该图形用户界面可以通过用户界面系统1702显示,其针对在图17中的命令队列管理器1726。在该例子中,用户可以浏览不同命令的状态。具体地,特定类型的命令可以通过窗口2500浏览。

[0256] 可以使用控件2502、2504和2506浏览命令。未决的命令可以通过选择控件2502浏览,已经执行的命令可以通过选择控件2504浏览并且出队列命令可以通过选择控件2506浏览。用户可以通过控件2508记录或改变命令在队列中的存储顺序。在该例子中,未决的命令已经被选择并且显示在窗口2500的区域2510中。

[0257] 现在参考图26,图26根据有利实施例描述浏览部件的用户界面的图示。窗口2600是通过用户界面系统1702呈现的图形用户界面的例子,其针对图17中的部件室1722。

[0258] 在该所描述的例子中,库中的软件飞行器部件可以被浏览。有效的部件可以通过控件2602的选择浏览,引入的部件可以通过控件2604的选择浏览,过期的部件可以通过控件2606的选择浏览,并且故障的部件可以通过控件2608的选择浏览。在该例子中,控件2602已经被选择并且位于库中的有效部件被显示在窗口2600的区域2610中。

[0259] 现在参考图27,图27根据有利实施例描写用于接收库中软件飞行器部件的过程流程图。图27中图解说明的过程可以在图17中的库1700中实现。具体地,这些过程可以在图17中库1700的管理组件1710中的部件室1722中实现。

[0260] 该流程开始于接收软件飞行器部件(操作2700)。在接收软件飞行器部件过程中也接收元数据以及用于软件飞行器部件的数据流。该流程确定软件飞行器部件的元数据是否正确(操作2702)。在这些例子中,元数据是与软件飞行器部件相关的数据和/或描述软件飞行器部件的数据。

[0261] 在这些例子中,下列元数据可以被检查:部件名是否符合规则表达 $[\sim/?:*?\"<>|.\backslash]*$ 并且是否少于或等于200字符;生产状态是否是BLACK_LABEL、RED_LABEL或BLUE_

LABEL;可应用的标准的长度是否大于或等于0并且小于或等于500字符;知识产权所有者的长度是否大于或等于0并且小于或等于100字符;版本日期是否是正确的日期格式;以及描述的长度是否大于或等于0并且小于或等于2000字符。

[0262] 如果部件的元数据是正确的,那么该流程将临时条目登记到库中的数据库(操作2704)。该临时条目用于提供接收部件的过程的状态。该条目最初指出已经开始接收部件。该流程也在文件系统中创建目录结构(操作2706)。当文件被接收时,该目录结构用于保存包含软件飞行器部件的文件的各部分或各段。

[0263] 判断是否已经完成软件飞行器部件的接收(操作2708)。如果还没有完成软件飞行器部件的接收,则在数据库中更新该进展(操作2710)并且将所接收的文件段保存在文件系统中(操作2712)。该进展可以显示在用户界面中。该流程接着返回到操作2708继续检查软件飞行器部件的接收操作的状态。

[0264] 当在操作2708中完成软件飞行器部件的接收时,该流程判断该部件是否完整(操作2714)。进行此操作以判断软件飞行器部件是否完整以及该部件是否有错误。该检查是通过匹配证书和接收到的部件来进行的。

[0265] 如果软件飞行器部件是完整的,则该流程将软件飞行器部件打包(操作2716)。该流程接着判断该打包操作是否成功(操作2718)。如果打包操作是成功的,则在数据库中将该部件标记为完整的(操作2720)。打包的文件被保存在文件系统中用于以后的检索(操作2722),此后该流程结束。

[0266] 再参考操作2718,如果打包操作不成功,则生成错误(操作2724)。此后,该流程将该条目从数据库中移除(操作2726)并且将已经保存的软件飞行器部件的数据移除(操作2728),此后该流程结束。再次参考操作2714,如果软件飞行器部件是不完整的,该流程也进行到操作2724。执行操作2726和2728以为软件飞行器部件接收失败清除数据库条目以及文件系统条目。

[0267] 再次参考操作2702,如果软件飞行器部件的元数据是不正确的,则该流程生成错误(操作2730),此后该流程结束。在操作2730和2724中生成的错误可以被存储在日志中用于以后使用。

[0268] 现在转向图28,图28根据有利实施例示出创建命令的过程流程图。图28中图解说明的过程可以在图17的库1700中实现。具体地,该过程可以在图17中的命令调度器1724中实现。

[0269] 该流程始于接收创建命令的用户请求(操作2800)。该命令可以通过例如由图17中的用户界面系统1702提供的用户界面接收。在这些例子中,用户可以选择三种命令类型中的一种。命令类型包括上行链路、下行链路和删除。该流程识别来自用户输入的命令类型(操作2802)。

[0270] 响应识别的命令类型,该流程生成参数和可能值的一个列表(操作2804)。该列表例如包括飞行器尾数、适用的装置名列表和接收命令的目的地外场可更换单元。该流程接着选择性地向用户呈现列表和值(操作2806)。在这些例子中,列表是上下文敏感列表,其根据用户之前做出的选择提供额外的选项或值。

[0271] 该流程接收从呈现的列表和值选择值的用户输入(操作2808)。该流程接着验证参数的上下文关系(操作2810)。在这些例子中,上下文敏感的值存在于图17中的用户界面

1702中。该界面实现命令类型中允许的目的地系统的值。操作2810在图17中的后端1704再次检查这些规则。后端1704可以服务于除图17中的用户界面1702以外的用户界面,该用户界面可以具有不同的验证规则。

[0272] 该流程创建一组命令(操作2812)。在操作2812中,该流程为命令类型、尾数和装置名的每个组合创建一个命令。当然,其他的规则和政策可以用于识别根据用户选择创建的命令。典型地,属于相同飞行器的相同类型的命令在逻辑上被分组。此后,命令组被保存在库的数据库中(操作2814),此后该流程结束。

[0273] 参考图29,图29根据有利实施例描述管理软件飞行器部件的过程的高级流程图。在这些例子中,图29中图解说明的过程可以在图17的库1700中实现。

[0274] 该流程开始于向分配装置发送命令结构(操作2900)。这些命令结构可以是,例如图21中的删除命令数据结构2100、图22中的上行链路命令数据结构2200或图23中的下行链路命令数据结构2300。这些命令结构响应来自于不同的装置的对命令的请求而被发送,该装置例如为代理服务器应用程序或软件维护工具。

[0275] 此后将命令文件发送到装置(操作2902)。在特定装置执行命令时,响应对命令本身的请求发送这些命令文件。取代发送命令文件,发送命令结构以减少在多个装置持续轮询时产生的通信总量。取而代之的是,在装置实际开始执行该命令时命令文件才被发送。此后,该流程发送软件飞行器部件(操作2904),此后该流程结束。在该操作中,软件飞行器部件作为命令执行的一部分发送。

[0276] 现在转向图30,图30根据有利实施例描写调度命令结构的过程流程图。图30中图解说明的过程是对图29中的操作2900更详细的描述。

[0277] 该流程开始于从装置接收请求(操作3000)。在这些例中,装置可以是,例如代理服务器应用程序或软件维护工具。当然该装置可以是任何与库联系或连接以获取命令的装置。

[0278] 该流程接着查询数据库,以获得相关联的命令或置于特定装置的命令队列中的命令(操作3002)。操作3002可以使用图17中的命令队列管理器1726实现。该流程接收查询的结果(操作3004)。

[0279] 此后,该流程创建包含该装置的命令的命令数据结构(操作3006)。该流程接着将命令数据结构返回给装置(操作3008),此后该流程结束。在这些例子中,基于装置对命令的请求创建命令数据结构。在其他实施例中,基于某事件或周期性事件(诸如计时器到期)来创建命令数据结构并且将命令数据结构广播到多个装置。

[0280] 现在转向图31,图31根据本发明有利实施例描写调度命令文件的过程流程图。图31中图解说明的过程是对图29中的操作2902更详细的描写。图31中图解说明的过程可以在组件中,例如图17中的命令调度器1724中实现。

[0281] 在该流程的这点上,装置已经接收要处理的命令数据结构。基于该命令数据结构,该装置可以进行该命令的某些处理。例如,装置可以开始建立与飞行器的通信链接。命令数据结构为装置提供了充足的信息以执行不同的过程。然而命令向飞行器的传播要求命令文件中的另外的信息。

[0282] 该流程开始于从装置接收对命令文件的请求(操作3100)。该流程查询数据库由该装置识别的命令(操作3102)。使用之前在命令结构中发送的唯一标识符进行该查询。

[0283] 该流程接着从数据库接收结果(操作3104)。在这些例子中,操作3102基于命令ID查询数据库并且检索关于该命令的所有信息,该命令用于创建打包版本的可扩展标记语言形式的命令。操作3104可能是多余的。这些结果用于创建命令文件(操作3106)。该流程将命令文件打包(操作3108)。此后,该流程将包返回给装置(操作3110),此后该流程结束。

[0284] 现在参考图32,图32根据本发明有利实施例描述调度部件的过程流程图。图32中图解说明的过程是对图29中的操作2904的更详细描述。在本例子中的过程可以使用图17中的命令调度器1724实现。

[0285] 该流程开始于从装置接收对软件飞行器部件的请求(操作3200)。该流程为软件飞行器部件查询数据库(操作3202)。该流程从文件系统检索打包的软件飞行器部件(操作3204),并从数据库检索软件飞行器部件的元数据(操作3206)。该流程接着对软件飞行器部件进行完整性校验(操作3208)。进行该完整性校验是为了确保软件飞行器部件在存储期间中没有被破坏。该完整性校验可以使用各种错误校验处理进行,包括哈希算法。

[0286] 基于完整性校验对软件飞行器部件是否有效做出判断(操作3210)。如果软件飞行器部件是有效的,那么将打包的软件飞行器部件返回到装置(操作3212),此后该流程结束。另一方面,如果软件飞行器部件是无效的,则返回错误消息(操作3214),此后该流程结束。

[0287] 现在转向图33,图33根据本发明有利实施例描述命令出队列的过程流程图。图33中图解说明的过程可以使用图17中的命令队列管理器1726执行。该过程被用于在命令已经被处理后将命令从数据库中的队列中移除。

[0288] 该流程开始于接收命令执行的通知(操作3300)。在该例子中,通知是从执行命令的装置接收的。该流程查找该命令以及其相关的组(操作3302)。该查找是使用命令的唯一标识符进行的。此外,与已经执行的命令相关的其他命令是冗余命令,这些命令可能已经被发送到同一飞行器的不同装置。

[0289] 该流程标记数据库的命令队列中的该命令并且使该命令出队列(操作3304)。该流程也使该组中的所有其他命令出队列(操作3306)。使其他命令出队列防止在今后将冗余的命令分派到不同的装置。此后,保存状态(操作3308),此后该流程结束。

[0290] 代理服务器

[0291] 现在转向图34,图34根据本发明有利实施例描述图解说明代理服务器应用程序中的数据流的图示。代理服务器应用程序3400与组件相互作用,组件例如为库3402、软件维护工具3404和机载电子分配系统3406。在这些例子中,地面组件3408提供用于在库3402和机载电子分配系统3406之间的信息传输。

[0292] 库3402可以向代理服务器应用程序3400发送新的命令和软件飞行器部件(消息A1)。处理这些命令的结果以及部件组作为命令状态信息可以由代理服务器应用程序3400返回给库3402(消息A2)。此外,代理服务器应用程序3400也可以向库3402发送下行链路以及事件日志文件(消息A3)。

[0293] 关于与机载电子分配系统3406传输信息,地面组件3408和代理服务器应用程序3400可以向机载电子分配系统3406发送新命令和软件飞行器部件(消息A4)。命令状态信息可以返回到地面组件3408,标识发送到机载电子分配系统3406的命令和部件的状态(消息A5)。此外,机载电子分配系统3406可以向地面组件3408发送下行链路文件(消息A6)。代理服务器3400是代理服务器应用程序(诸如图36中的代理服务器应用程序3606)的一个例子。

[0294] 代理服务器应用程序3400可以向软件维护工具3404发送新命令和部件(消息A7)。软件维护工具3404在处理这些文件后返回命令状态(消息A8)并且发送下行链路文件或事件日志(消息A9)。在这些例子中,软件维护工具3404可以与机载电子分配系统3406通信。软件维护工具3404提供与机载电子分配系统3406交换信息的备选路径。软件维护工具3404位于便携式数据处理系统中,该数据处理系统可以从与代理服务器应用程序3400相关的地址转移到机载电子分配系统3406所在的飞行器上。下面关于软件维护工具3404的描述对这些细节进行更详细的描述。

[0295] 尽管以特定的顺序描述了不同的交互,但是任何不同的消息和交互可以在任何时候同时发生。

[0296] 例如,在代理服务器应用程序3400向机载电子分配系统3406发送命令和飞行器部件的同时,机载电子分配系统3406可以将下行链路数据下载到代理服务器3400。此外,代理服务器应用程序3400可以同步地服务多个飞行器客户系统,例如软件维护工具3404和机载电子分配系统3406。

[0297] 现在参考图35,图35根据本发明有利实施例描述图解说明代理服务器应用程序的图示。代理服务器应用程序3500是图34中的代理服务器应用程序3400的实现的例子。在本例子中,代理服务器应用程序3500包括控制进程3502、数据库3504、文件系统3506、地面组件接口3508、软件维护工具接口3510和地面组件3512。在这些例子中,这两个接口可以使用应用程序接口(API)调用实现。

[0298] 数据库3504包括由控制进程3502处理的命令。在数据库3504中的每条记录都标识命令的状态。例如,记录可以标识命令是否已经被处理并且标识目标飞行器和飞行器上的目标外场可更换单元。在这些例子中,文件系统3506存储软件飞行器部件和下行链路数据。

[0299] 地面组件3512是在代理服务器应用程序3500中的软件组件,其与飞行器上的机载电子分配系统通信。地面组件应用程序接口3508提供控制进程3502用以与地面组件3512交换信息的调用。

[0300] 运行地面组件3512以允许任何进程,例如在代理服务器应用程序3500中的控制进程3502,与机载电子分配系统通信而不必为了与机载电子分配系统通信而进行特别的设计。结果,控制进程3502可以被改变或修改而不必包括用于与机载电子分配系统通信的协议。进一步地,可以进行对机载电子分配系统的改变不需要改变代理服务器应用程序3500中的所有进程。替代地,可以对地面组件3512进行修改或改变。

[0301] 软件维护工具应用程序接口3510提供的调用可以被控制进程3502用于与软件维护工具通信。数据库3504和文件系统3506的结构和组织可以与在软件飞行器部件维护环境的库中使用的数据库和文件系统的结构和组织相似。

[0302] 转到图36至39,图36至39根据本发明有利实施例描述图解说明在图35中的数据库3504中所使用的数据结构的图示。命令结果数据库表3600图解说明了命令结果的信息和记录。命令结果数据库表3600包括命令结果标识符3602、命令标识符3604、地面状态3606、飞行器状态3608、日期3610、命令类型3612、飞行器标识符3614和装置名3616。

[0303] 命令结果标识符3602唯一地标识特定的命令结果记录,并且命令标识符3604唯一地标识特定命令记录。在多个表中都可以发现命令标识符3604,以使表中的数据与特定的命令记录相关。地面状态3606标识命令状态消息的来源,在这些例子中,该命令状态消息可

以来自地面组件或机载电子分配系统。飞行器状态3608是源自于地面组件或机载电子分配系统的命令状态消息。地面状态标识被上传或下传的文件的上行链路或下行链路的状态。

[0304] 该信息提供文件的实际上行链路或下行链路完成的百分比。每个百分数可以被报告成独立的状态。以上行链路为例,完成四分之一的状态消息后接着是完成一半的状态消息,接着是完成四分之三的消息,最后是完成的状态消息,它们都作为发送到机载电子分配系统的文件的内容发送。每条消息的报告都指出适当数量的文件内容已经被成功地传送。同样的事情也发生在下行链路操作期间将文件内容写入到地面组件的情况。

[0305] 日期3610标识特定装置发送命令结果的日期。命令类型3612标识命令的类型,例如上行链路、下行链路或删除。飞行器标识符3614是标识在航空公司的机队中的特定飞行器的唯一值。装置名3616标识将命令结果发送到代理服务器应用程序的装置的名称。

[0306] 现在转向图37,图37根据本发明有利实施例描述下行链路文件数据库表的图示。在该例子中,下行链路文件数据库表3700图解说明下行链路文件数据库表中的信息。在该例子中,下行链路文件数据库表3700包括下行链路文件标识符3702、飞机标识符3704、装置3706、日期3708、文件名3710、文件通用资源定位器3712和文件状态3714。

[0307] 现在参考图38,图38根据本发明有利实施例描述命令和命令资源数据库表。在本例子中,命令数据库表3800表示命令而命令资源数据库表3802表示命令资源。命令数据库表3800包括命令标识符3804、飞机标识符3806、应用程序名3808、命令类型3810、装置名3812、系统名3814、日期3816、服务状态3818、优先级3820、命令组3822、打包的命令3824和打包的命令路径3826。

[0308] 命令资源数据库表3802包括:命令资源标识符3828、数据类型3830、应用程序标准3832、部件期满日期3834、所有者3836、名称3838、生产状态3840、发布日期3842、供应商3844、路径3846、包期满日期3848和命令标识符3850。在命令资源数据库表3802中标识的命令资源是在上行链路命令的包中的软件飞行器部件、下行链路命令的文件或配置报告和删除命令的文件或软件飞行器部件文件。

[0309] 命令标识符3804唯一地标识特定的命令结果记录。飞机标识符3806标识特定的飞行器。应用程序名3808标识外场可更换单元和飞行器。例如,应用程序名3808可以标识特定的外场可更换单元。装置名3812标识不同的装置,为该装置将命令分派给飞行器。装置名标识例如特定的代理服务器应用程序或软件维护工具。

[0310] 在这些例子中,名称可以是特定代理服务器应用程序或软件维护工具的特定名称。系统名3814标识应用程序存在于其上的系统的名称。日期3816标识库中的命令调度器创建该命令的日期。

[0311] 服务状态3818被用于标识命令的状态。该字段可以标识已经被成功地发送到机载电子分配系统的命令并且标识被软件维护工具报告为已经成功地上传到机载电子分配系统的命令。

[0312] 优先级3820是用来将队列中的命令排序以分配到机载电子分配系统的值。命令组3822可以用来为命令分组。打包的命令3824是包括打包格式的命令的文件的名称。打包的命令路径3826是标识存储有打包的命令的位置。

[0313] 在命令资源数据库表3802中,命令资源标识符3828唯一地标识命令资源记录。数据类型3830标识资源数据的类型。例如,数据类型可以是软件飞行器部件或文件。所有者

3836标识软件飞行器部件的知识产权所有者。在这些例子中,名称3838是文件或软件飞行器部件的名称。

[0314] 生产状态3840标识包中的软件飞行器部件的生产状态。该状态可以是,例如红标签、蓝标签或黑标签。红标签部件是处在工程开发中的不可交付的生产质量硬件或软件部件。蓝标签部件是被控制的和维护的,其仅限于在实验室环境下使用。黑标签部件被认为是生成完成的并且能够被交付给航空公司客户。

[0315] 发布日期3842标识包中的软件飞行器部件发布的日期。供应商3844标识软件飞行器部件的供应商。在这些例子中,路径3846标识软件飞行器部件的位置。例如,通用资源定位器字符串可以被用来检索部件。包期满日期3848是为包签名的证书期满的日期。命令标识符3850标识特定的飞行器命令记录。

[0316] 通过将文件名存储在打包的命令字段中并结合文件路径字符串,可以将打包的命令文件与命令表中的记录相关联。通过将文件保存在名称3838中并结合文件路径字符串,软件飞行器部件包文件可以与命令资源数据库表3802中的命令资源表中的记录相关联。

[0317] 在图39中,根据本发明有利实施例描述图解说明飞机命令数据库表的图示。在该例子中,飞机命令数据表3900提供为飞机命令找到的信息例子。飞机命令数据库表3900包括飞机命令标识符3902、消息标识符3904、飞机标识符3906、命令类型3908和命令XML 3910。

[0318] 飞机命令标识符3902被用于唯一地标识特定的飞行器命令记录。消息标识符3904是与特定下行链路命令相关的部分下行链路的标识符。为下行链路文件生成该标识符,所述下行链路文件不是发送到机载电子分配系统的下行链路命令的结果。命令XML 3910标识机载电子分配系统所发送的特定下行链路命令的可扩展标记语言文档文件格式,该命令XML将在机载电子分配系统请求部分下行链路文件时被检索。

[0319] 在这些例子中,不同表通过命令标识符彼此相关。不同的数据库表定义针对由代理服务器应用程序所处理的不同的数据元素。不同的过程可以使用这些表中的一个或多个,以指出何时插入、更新或删除记录。

[0320] 现在转向图40,图40根据本发明优选实施例描述代理服务器文件系统目录结构的图示。在该例子中,目录结构4000代表文件系统目录结构。目录结构4000是一种类型的目录结构的例子,该类型的目录结构可以在图35的文件系统3506中实现。目录结构4000可以标识存储在代理服务器应用程序上的文件系统上的不同类型的文件。

[0321] 在这些例子中,目录结构4000包括打包的命令4002、包4004、下行链路文件4006、下行链路文件档案4008、下行链路文件部分4010、存档的事件文件日志4012、事件日志4014和临时文件4016。在这些说明性例子中,该类型的目录结构被用于在文件系统中存储文件并且标识或定位文件系统中的文件。

[0322] 现在转向图41,图41根据本发明有利实施例描述从库接收信息的过程流程图。在该例子中,在图41中图解说明的过程可以在图35的控制进程3502中实现。该过程被用于从软件飞行器部件维护环境中的库接收命令和部件。响应一个事件而启动该过程。例如,该事件可以是计时器到期。在其他实施例中,事件可以由其他的源引起。例如,该事件可以通过用户输入启动。

[0323] 该流程开始于成功地标识已经执行的命令(操作4100)。这些命令是代理服务器应

用程序发送至一组飞行器的命令,在该组飞行器中成功地处理这些命令。命令可以是,例如删除软件飞行器文件、加载软件飞行器文件或从飞行器上的外场可更换单元下载数据。

[0324] 这些命令可以从代理服务器应用程序中的数据库(诸如图35中的数据库3504)识别。该特定的命令可以根据命令结果数据库表(诸如图36中的命令结果数据库表3600)中的命令标识符被识别。这些命令的标识形成发送给库的命令列表(操作4102)。操作4100和4102被用于发送命令状态信息给库。

[0325] 下一步,该流程向库请求命令列表(操作4104)。进行操作4104,以启动将新命令分配到一组飞行器的处理。该流程接收对请求的响应(操作4106)。判断是否进行响应而接收命令列表(操作4108)。如果命令列表没有被接收,该流程结束,没有新命令要处理。否则的话,该流程删除存储在数据库中、不在从库接收的新命令列表中的命令(操作4110)。

[0326] 在操作4110中,存在于具有代理服务器应用程序的数据库中的、不包括在从库检索的命令列表中的命令被认为是不必要由代理服务器应用程序处理或操作的命令。在这些例子中,该特征使得库成为假定在不同的代理服务器应用程序上处理的或存在的命令的权威源。

[0327] 如果代理服务器应用程序接收一个命令,并且该命令在处理前被用户删除了,那么库中的命令调度器为装置删除该命令。结果,代理服务器应用程序在以后的请求命令的周期中将不会接收命令列表中的该命令。通过这种方式,用户可以通过从命令队列屏幕删除代理服务器应用程序的未决命令为该特定代理服务器应用程序移除所有的命令。

[0328] 此后,该流程将新命令存储到数据库中(操作4112)。在这些例子中,命令列表可以采取命令数据结构的形式。代理服务器应用程序或者立即地或者在其他时间点上选择性地请求实际命令本身。

[0329] 该流程接着选择一条未处理的新命令进行处理(操作4114)。该流程请求包含该命令的包(操作4116)。响应该请求,该流程接收包(操作4118)。所接收的包接着被存储在文件系统中(操作4120)。该流程接着判断是否仍有未处理的新命令(操作4122)。如果还存在额外的未处理的新命令,则该流程返回到操作4114以选择另一个未处理的新命令进行处理。

[0330] 否则,在操作4122中该流程判断在接收到的新命令中是否存在上行链路命令(操作4124)。如果存在上行链路命令,那么选择未处理的上行链路命令进行处理(操作4126)。该流程接着判断包含软件飞行器部件的包是否已经存在于文件系统中(操作4128)。如果存在该包,那么该流程返回到如上所述的操作4124。

[0331] 如果不存在包,则该流程请求包含软件飞行器部件的包,该包对应来自库的命令(操作4130)。此后,该流程接收该包(操作4132)并且将该包存储在文件系统中(操作4134)。

[0332] 接着,该流程返回到操作4124以判断是否还存在额外的未处理的上行链路命令。如果不存在额外的未处理的上行链路命令,则流程结束。否则的话,该流程返回到操作4126,按照上面所描述的那样选择另一条未处理的上行链路命令。

[0333] 在执行图41中的流程期间,三种类型的事件日志消息被创建和记录。指出代理服务器应用程序成功地连接到库的记录是一个记录在日志中的事件。指出已经从库接收到所接收的命令列表的事件是另一个记录的事件。也为每条命令记录事件,事件置于由该命令识别的飞行器的队列中。成功发送到库的命令列表可以被用以辅助航空公司计划维护操作。

[0334] 现在转向图42,图42根据本发明有利实施例描述向库发送下行链路文件的过程流程图。图解说明在图42中的流程可以在控制进程(诸如图35中的控制进程3502)中实现。该流程图解说明在代理服务器应用程序向库发送从机载电子分配系统接收的下行链路文件时发生的不同操作。该流程可以由事件(诸如计时器)启动。本流程可以在和图解说明在图41中用于处理命令的流程不同的时间开始,以帮助展开通信网络和减少堵塞。

[0335] 该流程开始于识别数据库中未传输的下行链路数据的下行链路记录(操作4200)。判断在数据库中是否有未处理的记录(操作4202)。如果有未处理的记录,那么选择下行链路文件的未处理的记录进行处理(操作4204)。该流程向库发送包含下行链路数据的文件(操作4206)。

[0336] 此后,该流程将发送到库的文件存档(操作4208)。该流程接着将该文件的数据库记录更新为被存档的(操作4210)。接着判断是否还存在额外的未处理的记录(操作4212)。如果还存在额外的未处理的记录,则该流程返回到操作4204。

[0337] 否则,该流程识别数据库中比某选择的阈值更早的记录(操作4214)。该阈值可以是,例如从在指出下行链路文件被接收时间的的时间戳中的日期和/或时间开始的某个所选择的小时数。该流程从数据库删除任何被识别的记录(操作4216),此后该流程结束。再次参考操作4202,如果不存在未处理的记录,则该流程同样结束。

[0338] 现在转向图43,图43根据本发明有利实施例描述向库发送事件文件的过程流程图。图43中图解说明的流程可以在代理服务器应用程序组件(诸如图35中的控制进程3502)中实现。像其他流程一样,图43中图解说明的流程可以响应事件(诸如计时器)被启动。

[0339] 在本图中的流程支持将事件日志发送回库,用于分析在规划操作(诸如维护操作)中使用。发送到图43的库的事件日志获取事件消息,该事件消息是用户与应用程序用户接口系统的交互和/或应用程序组件之间的接口交互的结果。这种类型的信息可以在航空公司的业务处理期间被使用,用于在确保各种处理跟随特定操作的检查期间进行报告。

[0340] 该流程开始于比较当前日志文件和来自之前的处理循环的日志文件的副本(操作4300)。该流程标识任何从比较两个日志文件以后出现的任何新事件(操作4302)。该流程接着将在比较中发现的新条目的事件发送给库(操作4304)。发送到文件系统的文件的副本被存档(操作4306)。该流程接着发送任何延期日志文件给库(操作4308)。延期日志文件是从上一个时间段例如上一个日期开始就存在的文件。

[0341] 该流程将任何文件的副本存档在发送给库的文件系统中(操作4310)。该流程接着删除发送到库的延期日志文件(操作4312)。下一步,日志文件的上一个副本被删除并且当前的日志文件被设为副本用于下一次比较(操作4312)。该流程接着在事件日志目录内搜索装置名字目录(操作4314)。在操作4314中,具有装置名的子目录在代理服务器应用程序将从软件维护工具接收到文件的事件日志文件写入到文件系统时被创建。

[0342] 该流程在任何找到的子目录中搜索事件日志文件(操作4316)。然后,该流程将在子目录中发现的任何事件日志文件发送给库(操作4316)。该流程接着删除所有被发送的文件并清空子目录(操作4318)。此后该流程结束。

[0343] 现在转向图44,根据本发明有利实施例描述了发送信息给飞行器的过程流程图。在这些例子中,图44中图解说明的流程可以在软件组件中实现,该软件组件例如是图35中的代理服务器应用程序3500中的控制进程3502。在这些例子中,信息采取被发送到飞行器

上的机载电子分配系统的命令和软件飞行器部件的形式。

[0344] 该流程开始于从机载电子分配系统接收对下一条命令的请求(操作4400)。然后,该流程获取机载电子分配系统请求的下一条命令(操作4402)。在这些例子中,命令的实际文件被置于代理服务器应用程序的文件系统中。在数据库中的记录包括关于文件系统中的命令的元数据。

[0345] 该流程接着为命令更新数据库记录以指出该命令已经被服务(操作4404)。该流程判断下一条命令是否是删除命令(操作4406)。如果下一条要被处理的命令不是删除命令,则该流程接着判断当前飞行器是否在上传信息(操作4408)。

[0346] 如果飞行器当前正在上传信息,则该流程判断该飞行器是否也正在下传信息(操作4410)。如果该流程没有下传信息,则判断下一条命令是否是上行链路命令(操作4412)。如果下一条命令不是上行链路命令,则该流程寻找下一条命令并且将该命令返回给机载电子分配系统(操作4414),此后该流程结束。

[0347] 再次参考步骤4406,如果下一条被处理的命令是删除命令,则该流程进入到如上所述的操作4414。再次参考操作4412,如果下一条命令是上行链路命令,则该流程寻找并且返回飞行器的下一条下行链路命令或删除命令(操作4416),此后该流程结束。再次参考操作4410,如果飞行器正在下传,则该流程寻找并且返回飞行器的队列中的下一条删除命令(操作4418),此后该流程结束。

[0348] 返回参考操作4408,如果飞行器没有在进行上传,则判断该飞行器是否在下传(操作4420)。如果飞行器在操作4420中不进行下传,则该流程寻找并且返回飞行器的下一条命令(操作4422),此后该流程结束。如果飞行器在操作4420中正在下传,则判断飞行器的下一条命令是否是下行链路命令(操作4424)。

[0349] 如果下一条命令不是下行链路命令,则该流程进入到如上所述的操作4422。否则该流程寻找并且返回下一条上行链路命令或删除命令(操作4426),此后该流程结束。

[0350] 在这些例子中,执行判断发送哪条命令给飞行器的不同决定是为了避免在同一时刻向同一飞行器发送过多的上传和/或下行链路命令。这种类型的处理被用于改进或优化飞行器在和代理服务器应用程序通信时对带宽的使用。在该流程期间,事件日志消息被写入到日志文件,该消息报告何时软件飞行器部件被上传到飞行器。在其他有利的实施例中,其他类型的决定可以被用于实现其他期望的方针。例如,某些类型的命令可以被给予相对于其他类型的命令的优先权。所选类型的飞行器可以被给予相对于其他类型飞行器的优先级。

[0351] 下一步参考图45,根据本发明有利实施例描述接收软件飞行器部件的过程流程图。图45中图解说明的流程可以在软件组件中实现,例如图35中的控制进程3502。在本例子中说明的流程被用于从代理服务器应用程序获取软件飞行器部件。

[0352] 该流程开始于从机载电子分配系统接收对包含软件飞行器部件的包的请求(操作4500)。该流程定位对应该请求的包(操作4502)。接着该流程返回该包给机载电子分配系统(操作4504),此后该流程结束。

[0353] 现在转向图46,根据本发明有利实施例描述了从飞行器接收命令状态信息的过程流程图。图46中图解说明的流程可以在软件组件中实现,例如图35中的控制进程3502。该流程用于从飞行器上的机载电子分配系统接收状态信息。

[0354] 图46中的流程可用来获取关于在飞行器上的命令处理的状态信息。该状态信息可

以指出关于特定外场可更换单元的命令发送是否成功。另外,对于上行链路命令,状态也将指出软件飞行器部件是否被存储在飞行器上并且准备安装。在这些例子中,由机修工或技术人员启动在外场可更换单元上安装软件飞行器部件。在其他实施例中,这种类型的安装可以是自动的。

[0355] 该流程开始于从具有命令状态的机载电子分配系统接收调用(call)(操作4600)。该流程将具有来自命令状态的信息的新记录插入到命令结果数据库表中(操作4602),此后该流程结束。

[0356] 借助该信息,代理服务器应用程序可以将关于命令是否成功的状态信息发送回库。该信息允许标识软件飞行器部件何时存在在飞行器上并且准备安装在外场可更换单元上。在这些例子中创建三个事件日志消息。一个消息指出特定命令是否成功。消息也被发送回库,指出命令中哪些删除的文件被成功地删除。此外,标识失败的命令的标识也被记录到状态消息中。

[0357] 现在转向图47,根据本发明有利实施例描述了接收下行链路文件的过程流程图。图47中图解说明的流程可以在软件组件中实现,例如图35的控制进程3502。该流程图图解说明在下行链路文件从机载电子分配系统被发送到代理服务器应用程序时发生的过程。

[0358] 该流程开始于从机载电子分配系统接收调用以下载下行链路数据(操作4700)。在这些例子中,地面组件在文件写操作在之前被中断并且文件的完整内容没有被写入到文件时标识部分下行链路。如果成功地完成文件写操作,则该下行链路是完成的下行链路。如果该调用是为了发送下行链路的额外数据,则借助代理服务器应用程序,所接收的信息被添加到之前为文件系统上的下行链路存储的文件。

[0359] 该流程接着从下行链路文件接收数据(操作4702)。下一步,该流程判断该数据是否是针对部分下载的下行链路文件的数据(操作4704)。如果该数据是针对新下行链路文件的数据,则该流程创建下行链路文件(操作4706)。然后,该数据被存储在下行链路文件中(操作4708)。接着判断是否已为下行链路文件接收额外数据(操作4710)。如果已经接收额外数据,则该流程返回到操作4708。否则,该流程判断文件是否已完成(操作4712)。如果文件已完成,则该文件被存储在代理服务器上的文件系统中(操作4714),此后该流程结束。

[0360] 再次参考操作4712,如果文件没有完成,则该流程将文件标记为部分下载的下行链路文件(操作4716),此后该流程结束。再次参考操作4704,如果要下载的数据是针对部分下载的下行链路文件的数据,则该流程判断是否存在针对该数据的部分下行链路文件(操作4718)。

[0361] 如果存在下行链路文件,则该流程进入到如上所述的操作4708。否则,该流程发送错误给机载电子分配系统(操作4720),此后该流程结束。这个区域指出机载电子分配系统发送的数据的部分下行链路文件不存在于代理服务器上。在这种情况下,机载电子分配系统可在另一个数据传输中重新发送完整文件。

[0362] 在这些例子中,在机载电子分配系统接收来自代理服务器应用程序的命令的同时,机载电子分配系统可以发送下行链路文件给代理服务器应用程序。该流程考虑到在将数据下传到代理服务器应用程序期间可能会发生中断。如果下行链路数据的发送被中断,则成功写入的部分被保存供以后写入剩下的数据时使用。以这种方法,重新写入先前的数据是不必要的。在这些例子中,可以记录事件日志消息,该消息指出从特定飞行器的代理服

务器应用程序接收到下行链路数据。

[0363] 现在参考图48,根据本发明有利实施例描述了从软件维护工具接收状态信息的过程流程图。图48中图解说明的流程可以在图35中的控制进程3502中实现。该流程图解说明在从软件维护工具接收状态消息时发生的不同操作。

[0364] 该流程开始于从具有命令的命令状态信息的软件维护工具接收调用(操作4800)。然后,该流程将新记录插入到调用中标识的命令的命令结果数据库表中(操作4802)。该流程将记录标记为软件维护工具报告成功(操作4804)。

[0365] 该流程返回确认给软件维护工具(操作4806)。这些不同的消息可以与其他消息集合起来以传输回库。该流程接着将命令的本地副本标记为已发送到飞行器(操作4808),此后该流程结束。该流程防止代理服务器应用程序将命令重新发送回软件维护工具。

[0366] 现在转向图49,根据本发明有利实施例描述了向软件维护工具发送信息的过程流程图。图49中图解说明的流程可以在图35的控制进程3502中实现。该流程将消息以上行链路命令和软件飞行器部件的形式发送给软件维护工具。

[0367] 该流程开始于从软件管理工具接收对一系列命令的请求(操作4900)。该操作可以针对各种类型的命令。例如,该请求可以针对任何已经指派给特定软件维护工具的命令。该请求可以为特定飞行器、飞行器上的特定外场可更换单元或某标识符获取命令。

[0368] 作为对接收到该请求的响应,该流程在数据库中定位对应于该请求的命令(操作4902)。该流程接着从数据库接收结果(操作4904)。该流程将结果发送回软件维护工具(操作4906),此后该流程结束。软件维护工具可以使用与在图45中说明的用于将软件飞行器部件发送到机载电子分配系统的流程相似的流程请求包含软件飞行器部件的包。

[0369] 现在转向图50,根据本发明有利实施例描述发送软件飞行器部件列表给软件维护工具的过程流程图。该流程可以在软件组件中实现,例如图35中的控制进程3502。该流程可以被用于标识在代理服务器应用程序上可使用的软件飞行器部件。

[0370] 该流程开始于从软件维护工具接收对软件飞行器部件列表的请求(操作5000)。该流程接着向数据库发送查询,以识别存储在文件系统软件飞行器部件(操作5002)。从数据库接收结果(操作5004)。该软件飞行器部件列表被发送到软件维护工具(操作5006),此后该流程结束。

[0371] 在这些例子中,在操作5006中返回的列表可以包含代理服务器应用程序的目录中的唯一的软件飞行器部件名,即使该软件飞行器部件在该代理服务器应用程序上仅用于支持特别分派到该代理服务器应用程序而不是其他装置的命令。

[0372] 现在参考图51,根据本发明有利实施例描述了从软件维护工具接收下行链路文件的过程流程图。在该例子中图解说明的流程可以在代理服务器应用程序组件中实现,例如图35中的控制进程3502。

[0373] 该流程开始于从软件管理工具接收请求以下传文件(操作5100)。在操作5100中,该请求可以作为超文本传输协议请求。判断是否存在针对飞行器的目录(操作5102)。如果存在目录,则判断该文件是否已经存在于该目录中(操作5104)。

[0374] 如果该文件没有存在于该目录中,则该文件被写入到该飞行器的子目录中(操作5106)。然后,该流程在下行链路文件数据库表中为下载的文件插入新记录(操作5108),此后该流程结束。

[0375] 再次参考操作5104,如果该文件已经存在于该子目录中,则为文件名添加时间戳(操作5110),接着该流程进入到如上所述的操作5106。

[0376] 在操作5110中,时间戳被添加到文件名以允许同一文件的额外副本被写入而不覆盖或丢失原始文件。结果是,原始文件名连同额外文件一起存在,该额外文件具有除了添加的时间戳外都相似的文件名。文件的内容在某些情况下是相同的。再次参考操作5102,如果不存在飞行器的目录,则该流程为该飞行器创建一个子目录(操作5112)。该流程接着进入到如上所述的操作5106。

[0377] 在图52中,根据本发明有利实施例描述了从软件维护工具接收事件日志文件的过程流程图。在图52中图解说明的流程可以在图35的代理服务器应用程序3500中的控制进程3502中实现。

[0378] 该流程开始于从软件维护工具接收请求以将事件日志文件下载到代理服务器应用程序(操作5200)。该流程接着判断是否存在装置的子目录(操作5202)。在该例子中,该装置是软件维护工具。如果存在该装置的子目录,则为从软件维护工具接收的文件的文件名添加时间戳(操作5204)。该流程接着将该文件写入到装置名字目录(操作5206),此后该流程结束。

[0379] 再次参考操作5202,如果不存在该装置的子目录,则该流程为该装置创建子目录(操作5208)。该流程接着进入到如上所述的操作5204。

[0380] 软件维护工具

[0381] 在图3的软件飞行器部件管理设备300中,不同的有利实施例提供用于管理软件飞行器部件的计算机实现的方法、设备和计算机程序产品。不同的有利实施例提供位于便携式数据处理系统上的软件维护工具,该数据处理系统被用于通过地面网络建立与源的连接。通过该连接从源检索一组上行链路命令。通过建立的连接从源检索与上行链路命令相对应的一组软件飞行器部件以形成一组检索的软件飞行器部件。该组软件飞行器部件存储在便携式数据处理系统中。

[0382] 该便携式数据处理系统接着可以断开与地面网络的连接并且连接到飞行器中的飞行器网络。上行链路命令通过位于便携式数据处理系统中的地面组件从该组上行链路命令发给飞行器网络。所存储的对应于上行链路命令的软件飞行器部件接着可以通过地面组件被发送到飞行器网络。

[0383] 该软件维护工具可以被用于飞行器网络不能建立与地面网络的连接的情况下。例如,在某些机场,飞行器网络可能与存在的特定的地面网络不兼容。在其他例子中,地面网络中的失效或错误可以防止飞行器网络与地面网络通信以接收命令和软件飞行器部件。

[0384] 另外,在便携式数据处理系统上的软件维护工具也可以被用于从飞行器接收数据。该数据可以是,例如下行链路文件。

[0385] 下一步参考图53,根据本发明有利实施例描述图解说明数据流和软件维护工具的图示。软件维护工具5300与组件交互,组件例如为库5302、代理服务器应用程序5304和机载电子分配系统5306。这些组件也被称为源。在这些例子中,软件维护工具5300提供库5302和/或代理服务器应用程序5304与机载电子分配系统5306之间的信息的传输。

[0386] 库5302可以是,例如图3中的库304,而代理服务器应用程序5304可以是图3中的代理服务器应用程序306。机载电子分配系统5306可以是,例如图3中的机载电子分配系统

310。

[0387] 库5302发送新命令和部件给软件维护工具5300(消息E1)。在部件中处理这些命令的结果可以通过软件维护工具5300返回给库5302(消息E2)。此外,软件维护工具5300也可以返回下行链路和事件日志文件(消息E3)。

[0388] 依据特殊的实现或使用,软件维护工具5300可以通过代理服务器应用程序5304直接从库5302接收新命令和部件(消息E4)。以相似的方式,软件维护工具5300可以返回命令状态(E5)以及下行链路和事件日志文件(消息E6)给代理服务器应用程序5304,该代理服务器应用程序接着将该信息发送给库5302。

[0389] 关于与机载电子分配系统5306传输信息,软件维护工具5300可以发送新命令和软件飞行器部件给机载电子分配系统5306(消息E7)。软件维护工具5300可以从机载电子分配系统5306接收命令状态(消息E8)。该命令状态也可以包括发送给机载电子分配系统5306的软件飞行器部件的状态。机载电子分配系统5306可以发送下行链路和事件日志文件给软件维护工具5300用于传输给库5302(消息E9)。

[0390] 下面更详细地描述这些类型的传输的例子。另外,这些步骤和交互可以以特定的顺序发生并且任何不同的消息和交互可以在任何时间同时发生。例如,在软件维护工具5300从机载电子分配系统5306接收下行链路文件的同时,软件维护工具5300可向机载电子分配系统5306发送新命令和软件飞行器部件。在这些例子中,软件维护工具5300在便携式数据处理系统例如笔记本电脑上执行。图2中的数据处理系统200是用于实现笔记本电脑的数据处理系统的一个例子。

[0391] 软件维护工具5300可以从一个位置传送到另一个位置从而分配软件飞行器部件并且下载信息,例如从飞行器上的外场可更换单元下载数据或文件。在不同的有利实施例中,软件维护工具5300建立与机载电子分配系统5306的直接连接。在这些例子中,直接连接可以是有线连接或无线连接。这种类型的连接的建立不需要连接飞行器上的一个或多个数据处理系统与设置有软件维护工具5300的数据处理系统的网络。

[0392] 现在转向图54,根据本发明有利实施例描述了软件维护工具的框图。在本例子中,软件维护工具5400包括:库服务5402、数据库5404、文件系统5406、管理器5408、地面组件接口5410和地面组件5412。

[0393] 库服务5402提供与软件飞行器部件管理设备中的其他组件通信的接口。库服务5402为软件维护工具5400提供与组件,诸如库和代理服务器应用程序通信的接口。数据库5404包括信息,诸如关于文件系统5406中的一个或多个软件飞行器部件中的命令的元数据。

[0394] 此外,数据库5404也包含关于下行链路信息的信息。该信息以表或记录的形式存储在数据库5404中。另外,数据库5404可以存储从代理服务器应用程序接收的命令,以供飞行器中的飞行器数据处理系统上的机载电子分配系统执行。

[0395] 文件系统5406存储文件,诸如命令、软件飞行器部件和下行链路文件。根据特殊的实现,不同的文件可以被存储在文件系统5406中的包内。管理器5408包括用于管理软件维护工具5400的操作的进程。在这些例子中,管理器5408可以合并进程从而向用户呈现用户界面视图。这些视图为用户提供启动操作并且浏览信息的界面。

[0396] 地面组件接口5410提供到地面组件5412的接口。在这些例子中,地面组件接口

5410可以使用应用程序接口调用实现。地面组件5412与飞行器通信。在这些例子中,地面组件5412可以与位于飞行器中的飞行器数据处理系统上的机载电子分配系统通信。通过具有到地面组件5412的接口,可以为特定的飞行器或飞行器类型改变或修改地面组件5412而不影响软件维护工具5400中的其他组件。

[0397] 现在转向图55,根据本发明有利实施例描述命令和命令资源表的图示。在该例子中,命令表5500代表命令而命令资源表5502代表命令资源。这些表是在图54中的软件维护工具5400中的数据库5404中存在的表的例子。

[0398] 命令表5500包括:命令标识符5504、飞机标识符5506、系统名5508、应用程序名5510、命令类型5512、优先顺序5514、装置名5516、装置类型5518、日期5520、服务状态5522、执行状态5524、完成百分比5526、执行完成日期5528和命令资源列表5530。命令资源表5502包括:命令标识符5532、命令资源标识符5534、数据类型5536、包名5538、包路径5540、包文件大小5542、产品状态5544、应用程序标准5546、所有者5548、供应商5550、包期满日期5552和发布日期5554。

[0399] 为命令表5500和命令资源表5502说明的不同字段表示可能在软件维护工具数据库中的表中存在的字段。在命令表5500中,命令标识符5504唯一地标识特定的记录。命令标识符5504可以在各种表中存在,从而指向特定的命令记录。

[0400] 飞机标识符5506标识特定的飞机。在这些例子中,标识符可以唯一地标识在特定航空公司的飞行器机队中的飞行器。系统名5508标识在其上设置有软件飞行器部件的外场可更换单元的名字。系统名5508包括路径信息以标识外场可更换单元。日期5520标识在库中创建命令的日期和时间。服务状态5522标识已经成功发送到机载电子分配系统的命令并且标识已经报告给库的命令,该命令已经被成功地上或在机载电子分配系统上执行。

[0401] 执行状态5524提供命令是否已经在飞行器上执行的通知。具体地,该命令提供关于飞行器上的机载电子分配系统是否已经执行该命令的信息。在这些例子中,完成百分比5526指出由机载电子分配系统上传包中的软件飞行器部件的进展。执行完成日期5528标识何时完成命令的执行。命令资源列表5530标识包含关于包的信息的数据结构,诸如图55中的命令资源表5502。

[0402] 在命令资源表5502中,命令标识符5532与命令表5500中的命令标识符5504相似并且提供对特定命令记录的标识。命令资源标识符5534被用于标识数据库中特定的命令资源记录。数据类型5536标识资源的数据类型。例如,资源可以是软件飞行器部件或文件。在这些例子中,每条命令允许不同类型的信息与特定的命令关联。

[0403] 包名5538标识在其中置有软件飞行器部件的包的名字。包路径5540标识包含软件飞行器部件的包的位置。包文件大小5542标识包的大小。产品状态5544指出包含在包中的特定软件飞行器部件的产品状态。这些值可以是,例如,红标签、蓝标签或黑标签。

[0404] 应用程序标准5546在这些例子中标识软件飞行器部件的可应用的标准。所有者5548标识包含在包中的软件飞行器部件的任何知识产权的所有者。发布日期5554标识软件飞行器部件的发布日期。

[0405] 现在参考图56,根据本发明有利实施例描述了部分下行链路数据的图示。在该例子中,部分下行链路数据表5600是软件维护工具内的数据库中存在的表的例子,该数据库例如是图54中的数据库5406。如所描述的,部分下行链路表5600包含消息标识符5602、飞机

标识符5604、下行链路文件5606和部分文件XML 5608。

[0406] 消息标识符5602是与发送到机载电子分配系统的下行链路命令相关的部分下行链路的命令标识符。该标识符由飞行器上的机载电子分配系统为下行链路文件生成并且不是由库通过代理服务器应用程序或软件维护工具发送到机载电子分配系统的下行链路命令的结果。飞机标识符5604标识一组飞行器中的特定飞行器。

[0407] 下行链路文件5606指定到部分下传的文件完整目录路径。当机载电子分配系统请求下行链路文件(已经在之前的时间尝试下传该文件)时,软件管理工具返回至部分下传的文件的路径。

[0408] 部分文件XML 5608包含关于部分下传的文件的信息。该信息可以被机载电子分配系统使用,以从之前下传中断的地方恢复下传该被下传的文件。通过这种方法,文件的下传可以从其中断的地方开始,从而避免必须重新发送整个文件。

[0409] 接着转向图57,根据本发明有利实施例描述了下行链路表的图示。下行链路表5700是软件维护工具中数据库中的表的例子,该数据库例如是图54中的数据库5406。在这些例子中,下行链路表5700存储关于由机载电子分配系统发送的每个下传的文件的信息。下行链路表5700包括文件名5702、文件路径5704、飞机标识符5706、系统名5708、数据类型5710、文件大小5712、下传状态5714、已接收的下行链路5716,发送到库5718和发送到库的下行链路5720。

[0410] 文件名5702标识包含下行链路信息的文件的名称。在这些例子中,文件路径5704标识包含下行链路信息的文件的位置。飞机标识符5706标识飞行器,从该飞行器接收下行链路文件。该标识符是一组飞行器的唯一标识符,该组飞行器例如是特定航空公司的飞行器。该标识符可以是部件尾数。系统名5708标识在其上放置有软件飞行器部件的外场可更换单元的名称。数据类型5710标识数据的类型。对于下行链路信息,该数据被标识为文件。应用程序名5711标识在飞行器数据处理系统上的应用程序,该应用程序负责获取软件飞行器部件。

[0411] 文件大小5712标识包含下行链路数据的文件的大小。下行链路状态5714指示下行链路操作的状态。在这些例子中,下行链路状态5714示出成功的下行链路。在某些实施例中,部分下行链路可以由下行链路状态5714标识。库5718指出文件被下传到软件和维护工具的时间。发送到库5720的下行链路指出下传的文件被发送到库或代理服务器应用程序的时间。该信息被用于判断何时从软件维护工具删除已下传的文件。已下传的文件可以在下行链路文件被发送后一段可配置的时间后被删除,从而确保该已下传的文件在下行链路文件所发送到的库或代理服务器应用程序上备份。

[0412] 现在转向图58,根据本发明有利实施例描述了软件维护工具文件系统目录结构的图示。在该例子中,目录结构5800代表可以在文件系统例如图54中的文件系统5406中使用的文件系统目录结构。目录结构5800标识工作在软件维护工具上的文件系统中的不同类型的文件。在这些例子中,目录结构5800包括部件5802、下行链路5804、下行链路拆包目录5806、路径5808、应用程序5810、日志5812和配置5814。

[0413] 部件5802标识存储从库接收到的包的目录,该接收是直接从库接收或通过代理服务器应用程序接收。在这些例子中,包可以包括命令和/或软件飞行器部件。另外,包也可以从介质装载,例如从附连到在其上具有软件维护工具的笔记本电脑的闪存或硬盘驱动

器。该目录中的包可以被发送到飞行器上的机载电子分配系统。

[0414] 下行链路5804是用于存储从机载电子分配系统接收的下行链路文件和部分下行链路文件的目录。在这些例子中,下行链路文件可以通过飞行器的尾数组织,所述文件源自该飞行器。在这些例子中,下行链路5804可以包括以飞行器尾数命名的子目录。下行链路文件在这些例子中以打包的形式存储。已经被发送到库的下行链路文件不会自动地从下行链路5804删除。相反,这些文件在从其被发送至库或代理服务器应用程序后经过所选的时间之后被删除。

[0415] 下行链路拆包目录5806标识被软件维护工具用来将包的内容分开的目录。这些包被拆包,以提取关于下行链路文件的信息。未打包形式的文件可以使用文件的名字被存储在下行链路拆包目录5806内的目录中。

[0416] 路径5808标识包含SMT-route info.xml文件的目录。该文件包含系统、应用程序和由每个应用程序拣选的命令的列表。这些文件的内容被软件维护工具使用并且非直接地被库使用,从而确保上行链路命令被发送到适当的飞行器系统。

[0417] 应用程序5810标识其中安装软件维护工具的不同进程的目录。此外,与软件维护工具相关的日志也可以被存储在该目录下。这些日志包括,例如,在软件维护工具的操作期间记录的事件。

[0418] 日志5812是应用程序5810中的子目录并且在这些例子中包含最后被发送到库和/或代理服务器应用程序的事件logger.xml文件。配置5814是应用程序5808中的子目录并且包含属性文件,以定义软件维护工具的操作或行为,从而定义软件维护工具中不同组件的行为。

[0419] 下面转向图59,根据本发明有利实施例描述了图解说明在软件维护工具中所实现的界面组件的图示。在本例子中,用户界面组件5900是在图54的软件维护工具5400中的管理器5408中实现的用户界面组件的例子。用户界面组件5900包括连接视图5902、上行链路命令队列视图5904、上行链路本地清单(inventory)视图5906、下传文件视图5908、事件控制视图5910和从库检索视图5912。

[0420] 连接视图5902是用户界面组件,其提供一个区域以显示功能标签。在这些例子中,用户界面提供装置标识信息和下拉包,下拉包允许用户在不同的组件之间选择,组件例如有机载电子分配系统、库、代理服务器应用程序或其他装置。另外,该界面组件也可以提供将软件维护工具连接到从下拉包中选出的特定装置的控制件。

[0421] 上行链路命令队列视图5904提供浏览上传命令和部件的进展的界面。该视图具有删除命令和软件飞行器部件的控制件。上行链路本地清单视图5906提供用户界面以允许软件维护工具的操作员从介质载入或输入软件飞行器部件。该视图允许用户无需连接到地面网络而创建上行链路命令。用户可以选择上传到特定飞行器上的特定外场可更换单元的软件飞行器组件。该介质可以,例如,便携式介质,诸如闪存、便携式硬盘驱动器、光盘或数字通用光盘。下传文件视图5908提供用户界面以浏览从机载电子分配系统接收的下行链路文件。用户也可以使用该视图来删除下行链路文件和发出下行链路控制命令。

[0422] 事件控制视图5910允许用户浏览发生在软件维护工具的各种进程执行过程中的不同事件。例如,事件控制视图5910可以呈现发生在发送软件飞行器部件给机载电子分配系统期间的不同动作。这些事件可以包括,例如,连接到飞行器、发送文件和标识文件的成

功加载。

[0423] 从库检索视图5912是可以被用来启动从库或代理服务器应用程序获取命令和软件飞行器部件的过程的用户界面。图55中的命令表5500标识在命令表5500发现的字段。该视图往回发送成功上传的命令、下行链路文件和事件日志。

[0424] 下一步参考图60至65,这些图描述图59中用户界面组件5900的用户界面的示例实现。首先参考图60,窗口6000图解说明主屏幕或界面,其存在于图59中的用户界面组件5900中。具体地,窗口6000是图59中的连接视图5902的例子。窗口6000包括标签6002、6004、6006、6008和6010。这些标签可以被选择以给出软件维护工具中的不同功能和进程的控制和信息。

[0425] 区域6012显示关于软件维护工具的信息。在本例子中,区域6012指出软件维护工具连接到尾数标识的飞行器。列表6014提供软件维护工具与其建立连接的一列其他组件。控件6016允许用户启动与其他组件的连接。在这些例子中,用户可以选择各种组件,例如机载电子分配系统、库或来自一组代理服务器的代理服务器应用程序。

[0426] 现在参考图61,所选的标签6002在窗口6000中启动上行链路命令队列视图。在本例子中,该上行链路队列视图是由图59中的上行链路命令队列视图5904给出的用户界面的例子。在本例子中,区域6100以树形队列显示特定飞行器的命令。用户可以通过选择这些命令并且按下删除命令6102,从而从区域6100中删除一组命令。关于命令的状态信息在区域6103中给出。

[0427] 被显示的信息包括,例如,条目6104、期限6106、优先级6108、目的地系统6110、文件类型6112、命名6114、文件大小6116、上传状态6118和上传状态进展6120。条目6104标识特定的条目,例如软件飞行器部件名。期限6106是特定命令的期满日期。优先级6108标识命令被上传到飞行器上的目的地系统的顺序。目的地系统6110标识飞行器上的应用程序中的特定外场可更换单元,在其中部件被发送。类型6112标识包含在包中的条目的类型,诸如文件或软件飞行器部件。

[0428] 命名6114提供部件的简短的标识或描述。文件大小6116标识包含特定条目的包的大小。上传状态6118提供关于命令的处理、成功或失败的状态。上传状态进展6120提供图形进展条,显示特定命令完成的百分比。

[0429] 现在参考图62,根据本发明有利实施例描述了图解说明上行链路本地清单视图的用户界面的图示。在本例子中,标签6004已经被选择并且给出图59中的上行链路本地清单视图5906的用户界面。该特定的视图允许用户从本地源加载包或软件飞行器部件。这种类型的功能性允许用户在对库或代理服务器应用程序的访问是不可行的或被中断的情况下从另一源加载软件飞行器部件。此外,在库或代理服务器应用程序新部件中未找到的新部件或者更新的软件飞行器部件版本也可以以这种方法加载。软件飞行器部件或其他条目的本地清单可以在存储装置中找到,诸如硬盘驱动器、闪存、光盘或数字多功能光盘中。

[0430] 区域6200图解说明对本地清单的标识,该清单可以被加载到软件维护工具中。这些条目可包括软件飞行器部件和命令。特定的条目可以通过在区域6200中选择该条目并且按下从介质加载清单按钮6202来加载。在特定存储装置中找到的当前清单可以通过按下刷新清单6204而被刷新。

[0431] 关于区域6200中所选择条目的细节可以显示在区域6206中。在这些例子中,信息

可以包括,例如,清单条目6208、期满日期6210、飞机标识符6212、飞机目的地6214、类型6216、命名6218、文件大小6220、上传状态6222和上传状态进展6224。该信息与为软件飞行器部件显示的信息相似,该信息接收自库代理服务器应用程序,如同显示在图61的窗口6000中的信息。

[0432] 现在参考图63,根据本发明有利实施例描述下载文件视图的用户界面的图示。在该例子中,窗口6000显示用户界面组件的用户界面,诸如图59中的下传文件视图5908。该视图响应标签6006的选择而给出。在该用户界面中,关于从不同飞行器下传的数据的信息被显示在区域6300中。

[0433] 用户可以通过选择中止下行链路按钮6302中止或停止从飞行器上的机载电子分配系统的下传。当该按钮被选择,软件维护工具不再从飞行器接收任何额外的下行链路或信息,软件维护工具与该飞行器存在连接。下行链路可以通过按下所显示的恢复按钮被恢复。

[0434] 另外,用户可以通过选择重定向下行链路按钮6304将目的地是代理服务器应用程序的下行链路重定向到软件维护工具。该按钮的选择使得机载电子分配系统为目的地为代理服务器应用程序的所有下行链路信息重新选择路径,从而被发送到软件维护工具。如果软件维护工具与机载电子分配系统的连接是断开的,则该下行链路自动地发送到原始目的地。

[0435] 区域6306为在区域6300中给出的下行链路数据显示额外的信息。另外,用户可以在区域6306中浏览或删除下行链路文件。该删除可以通过选择特定的下行链路文件并且启动该删除命令来进行。

[0436] 在区域6306中给出的关于下传的文件的信息包括,例如,文件名6308、飞机标识符6310、系统6312、应用程序6314、数据类型6316、描述6318、文件大小6320、下传状态6322和下传日期和时间6324。文件名6308标识从飞行器下传或接收的文件的名称。飞机标识符6310标识从其处接收数据的飞行器。系统6312标识从其处接收数据的外场可更换单元。应用程序6314标识与数据相关的外场可更换单元上的软件飞行器部件。

[0437] 数据类型6316标识应用程序产生的数据的类型。在这些例子中,软件维护工具接收下传的文件,该文件具有表示未经请求(unsolicited)的下行链路的数据类型。描述6318在本例子中标识下传的文件名称。文件大小6320标识下传的文件的大小。下传状态6322标识数据是否成功地被下传到软件维护工具。下传日期和时间6324标识何时完成下传。该完成依据特定的情况可以是成功下传、失败下传或部分下传。

[0438] 现在转向图64,根据本发明有利实施例描述了图解说明事件控制的用户界面的图示。在本图示中,窗口6000显示了用户界面组件的用户界面,诸如图59中的事件控制5910。在所描述的例子中,本特定的视图响应选择标签6008而在窗口6000中给出。区域6400给出伴随特定软件维护工具时期发生的操作。在这些例子中,软件维护工具时期是一个时间段,软件维护工具在该时间段期间运行。图解说明在区域6400中的事件可以实时地给出。

[0439] 在这些例子中,可以通过选择保存事件控制按钮6402保存这些事件。通过选择清除事件控制按钮6404可以将事件从在区域6400中的显示中清除。此外,软件维护工具可以不需要用户干涉而自动地保存事件。在这些例子中,在区域6400中显示的每个条目包括日期和时间戳、执行特定操作的用户的用户标识符和标识操作已经被执行的消息。

[0440] 现在转向图65,根据本发明有利实施例描述了图解说明从库检索视图的用户界面的图示。在本例子中,窗口6000显示了图59中的从库检索视图5912的用户界面。响应对标签6010的选择给出该用户界面。该用户界面可以用于从库或代理服务器应用程序检索命令,以及浏览或看各种软件飞行器部件。此外,这就是丢包(loss)可以被用于创建命令以上传软件飞行器部件到机载电子分配系统的时候。

[0441] 在库中可获得的部件显示在区域6500中。为软件维护工具指派的特定软件飞行器部件可以通过按下执行库事务按钮6502被检索。用户也可以使用该界面创建上传软件飞行器部件给机载电子分配系统的命令。用户还可以从区域6500选择软件飞行器部件并且使用列表6504和列表6506指派特定飞行器和外场可更换单元。列表6504提供飞行器的标识。列表6506为软件飞行器部件标识在飞行器上的特定外场可更换单元。

[0442] 在进行完这些识别以后,可以通过按下从库中检索所选的清单按钮6508从库中检索软件飞行器部件。选择该按钮导致检索软件飞行器部件并且创建将软件飞行器部件上传到飞行器的命令。

[0443] 现在转向图66,根据本发明有利实施例描述了图解说明将命令和软件飞行器部件发送给飞行器时通过软件维护工具的数据流的图示。在本例子中为软件维护工具6600显示数据发送流,将命令和软件飞行器部件从库6602或代理服务器应用程序6604发送到飞行器上的机载电子分配系统(OBEDS)6606。软件维护工具6600执行的每一个不同步骤以及这些步骤的结果可以被记录为下载到库6602或代理服务器应用程序6604的事件。

[0444] 在本例子中,该流程开始于用户使用来自用户界面组件的用户界面启动与库6602或代理服务器应用程序6604的事务处理,该用户界面例如是图59中的从库检索视图5912。库服务6608检索已经成功地上传到机载电子分配系统6606的一列上行链路命令。库服务6608接着调用库6602或代理服务器应用程序6604并且传递成功地上传到机载电子分配系统6606的命令的命令标识符列表。库服务6608检索已经被成功地从命令表6610上传的上行链路命令列表。命令表6610是数据库6612中存在的表的例子。图55中的命令表5500标识在命令表6610中存在的字段。

[0445] 对于发送给库6602或的代理服务器应用程序6604的每一个命令标识符,相应的命令从数据库6612中的命令表6610中被删除。此外,库服务6608也发送来自文件系统6614的下行链路文件和事件日志。

[0446] 然后,库服务6608调用库6602或代理服务器应用程序6604以获取命令列表。这些命令与应该在队列中用于分配到不同飞行器的命令做比较。如果不在从库6602或代理服务器应用程序6604接收的命令列表中的命令存在于命令表6610中,那么这些命令从该表中被删除。但是软件维护工具6600的操作员生成的命令将被保留。在这些例子中,命令的删除针对之前从库6602或代理服务器应用程序6604发送的命令发生。

[0447] 对于每一个新接收的命令,库服务6608判断在文件系统6614中是否已经存在软件飞行器部件的包。如果不存在针对该命令的软件飞行器部件的包,那么库服务6608从库6602或代理服务器应用程序6604检索包含软件飞行器部件的包。任何检索到的包存储在文件系统6614中。如果成功地检索到包或包已经存在,那么新命令被置于数据库6612中的命令表6610中。如果成功地检索到包或包已经存在,则新命令被添加到上行链路命令队列管理器6617中的队列中。上行链路命令队列视图6618可以显示针对上行链路命令队列管理器

6617管理的命令的信息。

[0448] 然后,上行链路本地清单视图6616被更新或刷新。在本例子中,上行链路本地清单视图6616是用户界面组件,诸如显示在图62的窗口6000中的上行链路本地清单视图5906。该视图允许操作员看到存储在软件维护工具中的不同软件飞行器部件。通过知道哪些软件飞行器部件存在于文件系统6614中,操作员可以创建新命令以使用软件维护工具上传这些软件飞行器部件。此后,更新命令队列视图6618被更新。该视图可以是,例如,显示在图61的窗口6000中的上行链路命令队列视图5904。

[0449] 然后,操作员可以将软件维护工具6600从库6602或代理服务器应用程序6604断开连接。软件维护工具6600可以接着被传输到飞行器并且被连接到机载电子分配系统6606。在建立起该连接后,上行链路命令队列视图6618通过地面连接(OGC)接口6520将还没有成功地为特定的飞行器上传的所有命令自动地上传到机载电子分配系统6506。

[0450] 地面连接接口6620为地面组件(OGC)6622创建命令并且将该命令添加到地面组件6622的命令列表,用于一次检索一个。这些命令在上行链路命令队列管理器6617中被标识。

[0451] 当地面组件6622调用地面组件接口6620时,地面组件接口6620判断该飞行器是否已经在上传数据。如果该飞行器已经在上传数据,则返回空值给地面组件6622并且不改变命令列表中的命令。在这些例子中,地面组件6622与机载电子分配系统6606通信,从而确定飞行器是否在上传这些例子中的数据。

[0452] 如果该飞行器没有在上传,则命令队列中最早的上行链路命令被传递给地面组件6622。接着地面组件6622与机载电子分配系统6606通信,从而开始上传命令中识别的包。地面组件6622可以在软件飞行器部件的上传过程中获取状态信息。另外,地面组件接口6620可以更新上行链路命令队列视图6618,以显示进展条,诸如在图61中上传状态进展6120示出的进展条。

[0453] 当命令已经被成功地执行,上行链路命令队列视图6618更新命令表6610中的信息。此外,上行链路命令队列视图6618也更新命令表6610中命令字段的执行状态。

[0454] 现在转向图67,根据本发明有利实施例描述了图解说明在软件维护工具中处理下传的文件时的数据流的图示。在本例子中,软件维护工具6700可以接收下传的文件,该文件的下传由在连接到机载电子分配系统(OBEDS)6704的笔记本电脑上执行的应用程序6702启动。此外,来自外场可更换单元(LRU's)6706的未经请求的下行链路文件也可以被软件维护工具6700接收。当软件维护工具6700建立和机载电子分配系统6704的连接后,地面组件6708是提供与机载电子分配系统6704的通信的组件。

[0455] 地面组件(OGC)6708通过地面组件接口(OGC)6710与软件维护工具6700中的其他组件通信。在本例子中,下行链路文件被下传并且存储在文件系统6712中。当下行链路文件被传输到文件系统6712时,地面组件接口6710在数据库6716中的下行链路表6714中插入新记录。

[0456] 存储在文件系统6712中的不同的下行链路文件可以使用下传文件视图6718浏览。该视图是用户界面组件的例子,诸如图59中的下传文件视图5908。该视图可以被用来标识哪些下行链路文件已经被接收并且操作下行链路文件。图57中的下行链路表5700显示了可以在下行链路表6714中存在的字段的例子。

[0457] 然后,软件维护工具6700可以被移动并且建立与库6720或代理服务器应用程序

6722的连接。当建立该连接时,库服务6724标识还没有被发送给库6720或代理服务器应用程序6722的下行链路文件。对这些文件的标识可以在下行链路表6714中找到。

[0458] 在这些例子中,部分下行链路文件不发送给库6720或代理服务器应用程序6722。对于在下行链路表6714中标识的每一个下行链路文件,库服务6724都会确认这些文件仍旧存储在文件系统6712中。库服务6724接着将所有定位的下传的文件转发给库6720或代理服务器应用程序6722。发送给代理服务器应用程序6622的任何文件最后由代理服务器应用程序6722发送给库6720。

[0459] 在某些情况下,因为中断,文件可能只被部分地下传到软件维护工具。不同的有利实施例提供一种机制,通过该机制部分下传的文件被文件系统6612中的软件维护工具保存。这些部分下传的文件被保存并且下传文件的额外或剩余的部分可以在以后的时间取回并且添加到这些部分下传的文件中,以此形成完整的下行链路文件。通过这种方法,如果发生中断,则可以在其停止的地方重新开始数据的下传而不必再次下传完整的文件。

[0460] 现在转向图68,根据本发明有利实施例描述了图解说明数据流和由软件维护工具记录输入事件的图示。在该例子中,软件维护工具6800使用进程事件记录器6804将事件记录在文件系统6802中。进程事件记录器6804是在图54的管理器5408中找到的进程的例子。

[0461] 在这些例子中,进程事件记录器6804可以记录由软件维护工具6800在上传和下传数据时执行的所有不同步骤和这些步骤的结果。这种类型的信息可以显示在事件控制视图6806中,该事件控制视图是图59中的用户界面组件5900中的用户界面组件的例子。一个用户界面的例子是图64中的窗口6000。当软件维护工具6800通过库服务6812连接到库6808或代理服务器应用程序6810时,接收用户输入以传输数据并且存储在文件系统6802中的日志文件被转发给库6808和代理服务器应用程序6810。如果事件日志被成功地发送,则出于存档的目的,事件日志文件被重命名。

[0462] 现在转向图69,根据本发明有利实施例描述了图解说明软件维护工具中从库中检索部件时的数据流的图示。在本例子中,软件维护工具6900可以通过库服务6906连接到库6902或代理服务器应用程序6904。用户可以从库视图6908检索,从而标识存储在库6902和/或代理服务器应用程序6904上的部件。

[0463] 从库检索视图6908是从图59中的用户界面组件5900中的图59中从库检索视图5912的例子。图65中的窗口6000是该特定视图的用户界面的例子。部件可以被显示并且在从库检索视图6908中检索。用户可以选择一组部件并且从库6902和/或代理服务器应用程序6904检索这些部件,并且在文件系统6910中存储软件飞行器部件。这些部件接着被显示,以供用户创建一个或多个上行链路命令。

[0464] 现在转向图70,根据本发明有利实施例描述了图解说明软件维护工具在检索和创建命令期间的数据流的图示。在该例子中,软件维护工具7000可以检索部件并且使用从库检索视图7002创建命令。从库检索视图7002是用户界面组件的一个例子,例如图59中的从库检索视图5912,如同在图65的窗口6000中所给出的。

[0465] 当库服务7004连接到库7006或代理服务器应用程序7008,用户可以浏览从库检索视图7002检索出的部件列表。用户可以通过该视图选择部件,并且通过库服务7004启动这些部件的下传。库服务7004检索出的部件被存储在文件系统7010中。在这些例子中,软件飞行器部件被存储为包。上行链路本地清单视图7012可以被刷新。

[0466] 利用从库检索视图7002,用户可以创建命令,所述命令被存储在数据库7018中的命令表7014中。这些命令可以被添加到上行链路命令队列管理器7020中,由地面组件(OGC)7022通过到机载电子分配系统(OBEDS)7036的地面组件(OGC)接口7024执行这些命令。上行链路命令队列管理器7020是图54中的软件维护工具5400中的管理器5408中的组件的例子。该过程的状态可以通过上行链路命令队列视图6928浏览。

[0467] 现在参考图71,根据本发明有利实施例描述了图解说明从替代源上载软件飞行器部件的图示。在本例子中,软件维护工具7100可以通过上行链路本地清单管理器7106将软件飞行器部件从介质7102上载至文件系统7104中。该视图是图59中的上行链路本地清单视图5906的例子。本视图使用图形用户界面,例如图62中的窗口6000。

[0468] 对这种从介质7102上传或下传过程的控制可以使用上行链路本地清单视图7108进行。软件飞行器部件可以从不同于库或软件代理服务器应用程序的其他源上载到软件维护工具7100中。通过允许这种类型的灵活性,软件维护工具7100就可以允许在到库或代理服务器应用程序的连接不可用的情况下最近的部件或从正常源还不可获得的新部件被上载至飞行器。

[0469] 现在转向图72,根据本发明有利实施例描述了用于管理软件飞行器部件的过程的高级流程图。图72中图解说明的流程可以在软件维护工具中实现,诸如图54中的软件维护工具5400。

[0470] 该流程开始于通过地面网络建立便携式数据处理系统和源之间的连接,从而形成建立的连接(操作7200)。然后,该流程通过建立的连接从源检索一组上行链路命令(操作7202)。该源可以是,例如,代理服务器应用程序、库或甚至是本地存储装置。

[0471] 该流程接着通过建立的连接从源检索与该组上行链路命令相应的一组飞行器部件以形成一组检索到的软件飞行器部件(操作7204)。该流程将该组检索到的软件飞行器部件存储在便携式数据处理系统中以形成一组存储的软件飞行器部件(操作7206)。

[0472] 该流程接着断开具有所存储软件飞行器部件的便携式数据处理系统与地面网络的连接(操作7208)。在这些例子中,便携式数据处理系统被转移到一个位置,以允许便携的数据处理系统连接到飞行器上的飞行器网络。下一步,该流程将具有所存储飞行器部件的便携式数据处理系统连接到飞行器上的飞行器网络(操作7210)。

[0473] 该流程接着通过便携式数据处理系统中的地面组件从该组上行链路命令发出上行链路命令给飞行器数据处理系统(操作7212)。该流程通过地面组件向飞行器数据处理系统发送与上行链路命令相对应的所存储软件飞行器部件组中的所存储的飞行器部件(操作7114),此后该流程结束。

[0474] 现在转向图73,根据本发明有利实施例描述用于管理软件飞行器部件的过程的更详细流程图。图解说明在图73中的流程可以在软件维护工具中实现,诸如图54中的软件维护工具5400。在本图中的流程说明在连接到源时软件维护工具中发生的不同步骤,该源诸如为库或代理服务器应用程序。

[0475] 该流程开始于接收进行事务处理的请求(操作7300)。在本例子中,进行事务处理的过程可以由用户通过图59中的用户界面组件5900中的用户界面启动。具体地,该过程可以由用户通过用户界面将用户输入键入到图59中的从库检索视图5912而被启动,该用户界面诸如为图解说明在图65中的窗口6000。

[0476] 该流程接着检索发送到机载电子分配系统的一系列上行链路命令(操作7302)。在本例子中,该列上行链路命令是其中上行链路命令识别的软件飞行器部件已经被成功地发送到机载电子分配系统的那些上行链路命令。这些不同命令可以被存储在数据库中的表中,诸如图55中的命令表5500。图55中的命令表5500内的每个记录可包括命令是否成功发送的指示。

[0477] 然后,该流程调用源(操作7304)。该源可以是,例如,库或代理服务器应用程序。该流程发送这些命令列表给源(操作7306)。发送给源的命令接着从数据库和软件维护工具被删除(操作7308)。

[0478] 该流程接着调用源以检索新命令(操作7310)。一系列命令从源接收(操作7312)。在操作7312中,不同于代理服务器或应用程序从库接收命令的方式,命令以未打包的形式被接收。该流程接着从数据库删除不在列表中的命令(操作7314)。结果,源是权威或者提供关于哪些命令由软件维护工具执行的代理。

[0479] 如果用户期望移除或删除在飞行器上执行的命令,那么这些命令可以在源处被删除。发送到软件维护工具的该列命令导致任何不在该列表中的命令被删除。因此,这种类型的流程允许更新在软件维护工具上被执行的命令。

[0480] 该流程选择未处理的命令进行处理(操作7316)。判断是否存在针对该命令的包含软件飞行器部件的包(操作7318)。在操作7318中,该流程检查软件维护工具上的文件系统以判断包含软件飞行器部件的包是否已经存在于文件系统中。如果包不存在,则该流程检索该包(操作7320)。

[0481] 下一步,判断是否存在额外的未处理的命令(操作7322)。如果存在额外的未处理的命令,该流程返回操作7316。如果存在针对该命令的包,该流程从操作7318进行到操作7322。该流程接着将命令添加到队列(操作7324)。该流程然后更新软件飞行器部件的清单(操作7326)。

[0482] 图74说明了当便携式数据处理系统连接到飞行器网络时,发生在软件维护工具中的操作,该软件维护工具位于该便携式数据处理系统上。

[0483] 在这些例子中,软件维护工具可以用于发送软件飞行器部件给在飞行器网络中的飞行器数据处理系统上运行的机载电子分配系统。在这些例子中,队列可以是,例如图65中的上行链路命令队列管理器6517中的队列。该流程接着更新软件飞行器部件的清单(操作7326),此后该流程结束。

[0484] 现在转向图74,根据本发明有利实施例描述从软件维护工具发送软件飞行器部件给机载电子分配系统的过程流程图。在本例子中,该流程可以在软件维护工具中实现,诸如图54中的软件维护工具5400。该流程开始于探测与飞行器数据处理系统上的机载电子分配系统的连接(操作7400)。

[0485] 该流程判断在命令队列中是否存在命令(操作7402)。如果存在命令,则该流程判断飞行器是否正在上传数据(操作7404)。如果飞行器当前没有在上传数据,则该流程发送请求给机载电子分配系统,以上传包含软件飞行器部件的包(操作7406)。该流程接着获取上传的状态(7408)。该状态可以显示在用户界面上,诸如图61中的窗口6000。操作7408在继续上传包时进行。

[0486] 在上传完成以后,判断是否已经成功上传具有软件飞行器部件的包(操作7410)。

如果成功上传包,则更新命令表(操作7412)。在这些例子中,该表是命令表,诸如图55中的命令表5500。该流程接着返回到操作7302,判断队列中是否存在需要处理的额外命令。

[0487] 再次参考操作7410,如果软件飞行器部件的上传不成功,那么生成错误(操作7416)并且该流程返回到如上所述的操作7402。再次参考操作7404,如果飞行器正在上传数据,则返回空值(操作7414),此后该流程结束。

[0488] 现在参考图75,根据本发明有利实施例描述接收下行链路数据的过程流程图。在图75中图解说明的流程可以在数据软件维护工具中实现,诸如图54中的软件维护工具5400。

[0489] 图75中的流程开始于从机载电子分配系统接收调用以检索部分下行链路文件(操作7500)。判断部分下行链路文件是否包含在部分下行链路表中(操作7502)。该部分下行链路表可以是表,诸如,图56中的部分下行链路表5600。如果在表中没有找到该部分下行链路文件,则该流程接收调用,以从机载电子分配系统获取下行链路文件的名称(操作7504)。

[0490] 下一步判断是否存在足够的空间以存储下行链路文件(操作7506)。如果存在足够的空间,则下行链路文件创建在名为“downlinks\”的目录下并且文件名称被返回给机载电子分配系统(操作7508)。该流程接着将下行链路数据存储到“downlinks\”目录下的下行链路文件中(操作7510)。

[0491] 接着判断是否成功地存储下行链路文件(操作7512)。如果所有的下行链路数据都成功地存储,则该流程将下行链路文件添加到下行链路数据库表(操作7514)。该表可以是诸如图57中的下行链路表5700。该流程接着更新下行链路文件视图以显示新文件(操作7516)。该视图例如是在图63的窗口6000中给出的下传文件视图5908。

[0492] 该流程接着判断数据是否被写入到部分下行链路文件中(操作7518)。如果数据没有被写入到部分下行链路文件中,该流程结束。否则,在部分下行链路表中的部分下行链路记录被删除(操作7520),此后该流程结束。在这种情况下,用剩下的下行链路数据完成了部分下行链路文件,并且不再需要该部分下行链路文件的标识。

[0493] 再次参考操作7512,如果对下行链路文件的所有数据的存储是不成功的,则该流程接收来自机载电子分配系统的调用,以存储部分下行链路文件(操作7522)。在这种情况下,机载电子分配系统可以以多种理由中断下传数据。例如,可用带宽量不足以下传数据和上传其他信息。该流程接着在部分下行链路数据库表中创建记录(操作7524),此后该流程结束。

[0494] 再次参考操作7506,如果不存在用于下行链路文件的足够空间,则返回空值给机载电子分配系统以指出不存在用于下行链路数据的足够空间(操作7526)。往回参考操作7502,如果部分下行链路文件存在于部分下行链路表中,则该流程返回部分下行链路文件信息给机载电子分配系统(操作7528)。该信息包括发送下行链路文件的剩余下行链路数据的起始点或偏移。该流程接着进行到如上所述的操作7510。

[0495] 因此,在这些有利实施例中描述的软件维护工具提供从库向飞行器数据处理系统传输软件飞行器部件的额外特征。在不同的有利实施例中,软件维护工具可以连接到地面上的库或代理服务器应用程序,从而接收命令和软件飞行器部件。该软件维护工具接着可以从地面网络断开连接并且在物理上转移到一个位置,用于连接到飞行器网络。在这个位置上,软件维护工具连接到飞行器网络并且传输软件飞行器部件和命令给机载电子分

配系统,该机载电子分配系统在飞行器中的飞行器网络上的数据处理系统上运行。

[0496] 此外,软件维护工具允许操作员使用软件维护工具中的视图组件给出的图形用户界面独立于库创建命令。该软件维护工具也包括这样的特征,其允许该组件从库或代理服务器应用程序以外的其他源接收软件飞行器部件。

[0497] 机载电子分配系统

[0498] 不同的有利实施例也给出用于与飞行器传输信息的计算机实现的方法、设备和计算机程序产品。在一个有利实施例中,计算机实现的方法被用于与飞行器传输信息。在运行在飞行器中的飞行器数据处理系统中的机载电子分配系统和地面组件之间建立连接。

[0499] 在这些例子中,地面组件可以位于软件应用程序中的地面网络中,例如软件维护工具或代理服务器应用程序。响应从机载电子分配系统通过连接发出的对命令的请求,识别由机载电子分配系统执行的命令。该识别的命令被从地面组件发送给机载电子分配系统。事务处理标识符被分配给该命令。

[0500] 和命令关联的事务处理的状态在机载电子分配系统上和地面组件上通过使用事务处理标识符保持。上传由机载电子分配系统启动。软件飞行器部件接着从地面组件发送到机载电子分配系统以进行上传。该传输的状态被存储。

[0501] 现在转向图76,根据本发明有利实施例描述了用于与飞行器传输信息的组件图示。机载电子分配系统7600是机载电子分配系统的例子,诸如图3中的软件飞行器部件管理设备300中的机载电子分配系统310。

[0502] 在本说明性例子中,机载电子分配系统7600和大容量存储器7602是位于飞行器网络中的飞行器数据处理系统上的组件。机载电子分配系统7600是图1中的机载电子分配系统146的例子。大容量存储器7602是图1中存储装置148的例子。这些组件是飞行器数据处理系统的部件,诸如飞行器网络101中的飞行器数据处理系统144的部件。

[0503] 地面组件7604和地面组件接口7606是可以在代理服务器应用程序或软件维护工具中存在的组件例子,诸如图35中的代理服务器应用程序3500或图54中的软件维护工具5400。在这些例子中,地面组件7604和机载电子分配系统7600可以交换信息。命令7607、软件飞行器部件7608、下行链路文件7610和状态7612是可以用机载电子分配系统7600传输的信息的例子。

[0504] 在这些例子中,地面组件7604可以发送命令7607给机载电子分配系统7600。机载电子分配系统7600可以执行该命令以进行事务处理。该事务处理可以是,例如数据上传或下传。上传包括发送软件飞行器部件7608给机载电子分配系统7600。下传包括发送下行链路文件7610给地面组件7604。

[0505] 另外,在这些例子中,不同事务处理的状态同时由地面组件7604和机载电子分配系统7600保持。状态7612由机载电子分配系统7600发送至地面组件7604以提供通过执行命令进行的特定事务处理的状态,该命令诸如是命令7606。这个状态通过命令标识符与特定命令或事务处理相关联。

[0506] 软件飞行器部件7608可以被发送给机载电子分配系统7600用于与软件飞行器部件7614一起存储在大容量存储器7602中。下行链路文件7610可以是来自于大容量存储器7602的下行链路文件7616的下行链路文件。

[0507] 状态信息7618可以被存储在大容量存储器7602中并且包括状态信息,诸如状态

7612。状态信息7618可以指出特定的软件飞行器部件已经被成功地存储在大容量存储器7602中的软件飞行器部件7614中。状态信息7618允许在软件飞行器部件被识别为已经由机载电子分配系统7600成功地上传并且存储在大容量存储器7602中时启动将软件飞行器部件从大容量存储器7602加载到外场可更换单元。

[0508] 此外,状态信息7618可以识别下行链路文件诸如下行链路文件7610是否已经成功地被下传。如果发生下行链路文件7610的部分下传,那么状态信息7618提供下行链路文件7610中的什么信息已经被传输的状态。因此,维持多少信息已经被下传到地面组件7604的状态可以被用于在以后的时间点下传下行链路文件7610的剩余信息而无需重新开始下行链路文件7610的完整传输。

[0509] 地面组件接口7606提供其他组件与地面组件7604的接口。以这种方式,地面组件7604可以是可互换的或修改为地面组件的其他版本或配置,从而提供对特定机载电子分配系统的接入,该机载电子分配系统可以具有用于交换信息或处理命令的不同的协议。在这些例子中,地面组件7604包含与机载电子分配系统7600传输信息所需的进程。如果使用与地面组件7604不兼容的不同的机载电子分配系统,则地面组件7604可以被替代为另一地面组件。

[0510] 因此,在地面网络中的其他软件组件不是必须被改变,例如,代理服务器应用程序和软件维护工具中的其他组件不要求更改为能够与机载电子分配系统通信。

[0511] 现在转向图77,根据本发明有利实施例描述了图解说明用于针对命令轮询的消息流的消息流图示。在本例子中,在该消息流中包含的组件为:地面组件(OGC)接口7700、地面组件7702和机载电子分配系统7704。

[0512] 在本例子中,机载电子分配系统7704针对命令轮询地面组件7702(消息T1)。响应于被轮询,地面组件7702发送获取命令请求给地面组件接口7700(消息T2)。本命令被地面组件接口7700用于识别位于机载电子分配系统7704的代理服务器应用程序或软件维护工具中的命令。

[0513] 作为响应,命令或指向打包的命令文件的指针被返回给地面组件7702(消息T3)。在这些例子中,代理服务器应用程序返回指向包含该命令的打包的文件的指针,诸如通用资源定位器。借助软件维护工具,实际的命令本身在消息T3中被返回。如果命令不存在,则在消息T3中返回空值或某个其他指示。返回的命令接着被发送给机载电子分配系统7704(消息T4)。机载电子分配系统7704可以接着处理并且执行在消息T4中接收的命令。

[0514] 现在转向图78,根据本发明有利实施例描述了图解说明发送状态信息的消息流的图示。在本例子中,在消息流中的组件包括地面组件接口7700、地面组件7702和机载电子分配系统7704。机载电子分配系统7704为由机载电子分配系统7704执行的各种操作和处理提供状态信息。该状态信息可以包括,例如,已经上传的软件飞行器部件的状态、下行链路文件的状态和/或其他合适的信息。

[0515] 机载电子分配系统7704发送状态给地面组件7702(消息U1)。该状态由地面组件7702转发给地面组件接口7700(消息U2)。在这些例子中,该状态信息可以接着由代理服务器应用程序或软件维护工具处理

[0516] 存在下传数据的两个阶段。图79说明了第一个阶段,在该阶段中作出对下传数据的请求,并且图80描述了第二个阶段,在该阶段中数据被下传。现在参考图79,根据本发明

有利实施例描述了请求下载数据的消息流图示。图79中的消息流显示了下载数据的第一个阶段。在这些例子中，图79显示了请求下载数据。如根据下面的图80描述的，第二个阶段用于实际传输下行链路数据。

[0517] 在本例子中，第一个阶段具有两种情况。在情况7902中，在部分下行链路可用的情况下请求下载信息。

[0518] 在情况7900中，机载电子分配系统7706发送下载文件的请求（消息V1）。在消息V1中，如果没有存储下行链路文件的空间，则拒绝该请求。作为响应，地面组件7702发送请求以判断部分下行链路记录是否被提供给地面组件接口7700（消息V2）。作为响应，地面组件接口7700发送请求以获取与该请求相关的部分下行链路，从而发送消息V1（消息V2）。在消息V2中发送的请求包括飞机标识符和下行链路标识符。本信息被地面组件接口7700用于判断是否存在此特定下行链路文件的部分下行链路文件。

[0519] 地面组件接口7700返回空值给地面组件7702，指出不存在针对所请求的下行链路的部分下行链路文件（消息V3）。作为响应，地面组件7702做出下载下行链路文件的请求（消息V4）。在这些例子中，消息V4中的消息是下载整个文件的请求。在这些例子中，t消息包括关于文件大小的信息。如果空间可用，则地面组件接口7700返回位置，从而将文件下载至地面组件7702（消息V5）。如果空间不可用，则返回空值给消息V5。

[0520] 作为响应，地面组件7702返回响应给机载电子分配系统7704（消息V6）。该消息为可以进行下载的指示或者拒绝请求。

[0521] 在第一个阶段的情况7902中，机载电子分配系统7704请求下载下行链路文件的部分文件（消息V7）。作为响应，地面组件7702请求判断是否已经存在针对所请求的下行链路的部分下载的文件（消息V8）。

[0522] 响应接收该消息，地面组件接口7900给地面组件7702返回包含对存在的部分下载的文件引用的文档（消息V9）。在这些例子中，根据特殊的实现，该文档是可扩展标记语言（XML）文档并且引用可以是指针或通用资源定位符（URL）。

[0523] 当返回引用时，地面组件7702向机载电子分配系统7704发送对下载部分下行链路文件的请求的响应（消息V10）。该响应在本例子中包括可以继续下载的指示和使用的偏移值。该偏移值标识应该从下行链路文件的何处开始下载数据。该偏移值是从已经为下行链路文件接收的下行链路信息中识别的。

[0524] 现在转向图80，根据本发明有利实施例描述下载数据的消息流图示。像图79一样，该下载过程包括两种情况，情况8000和情况8002。情况8000涉及不用部分下行链路下载数据并且情况8002涉及用部分下行链路下载数据。在图79中，情况7900说明部分下行链路不可用的情况，而情况7902说明了在地面组件上部分下行链路文件可用的情况。

[0525] 在情况8000中，消息流始于机载电子分配系统7704下载下行链路文件给地面组件7702（消息W1）。地面组件7704请求将文件自机载电子分配系统7704下载到地面组件7700（消息W2）。该消息包括文件大小和其他合适的下行链路信息。

[0526] 地面组件接口7700返回响应给地面组件7702（消息W3）。如果没有空间可用于下载下行链路文件则返回空值。如果下行链路文件可以被下载，则地面组件7702将信息写入到文件并且返回响应给机载电子分配系统7704（消息W4）。然后地面组件7702向地面组件接口7700发出请求以存储文件（消息W5）。

[0527] 下一步,在情况8002中,机载电子分配系统7704下传文件到地面组件7702(消息W6)。然后地面组件7702从地面组件接口7700请求部分下行链路文件(消息W7)。在本例子中,文件由地面组件接口7700返回给地面组件7702(消息W8)。

[0528] 这时,地面组件7702将信息写入到文件以完成下行链路文件并且返回响应给机载电子分配系统7704(消息W9)。在本例子中,写入到文件的位的数量标识在响应中。然后,地面组件7704发送请求给地面组件接口7700,以存储下传的文件(消息W10)。

[0529] 响应该消息,地面组件接口7700可以将文件存储到地面组件的文件系统中。地面组件可以是存储在代理服务器应用程序或软件维护工具中的文件。

[0530] 现在参考图81,根据本发明有利实施例描述了图解说明在文件仅仅被部分地传输时的消息流的图示。在本例子中,机载电子分配系统7704使用正常下传序列下传文件,在该下传序列中连接失败或停止(消息X1)。响应于只接收部分文件,地面组件7702发送请求给地面组件接口7700以存储部分下传的文件(消息X2)。响应于接收该请求,部分下行链路文件由地面组件接口7700存储在文件系统中。该文件系统可以位于代理服务器应用程序或软件维护工具中。

[0531] 现在转向图82,根据本发明有利实施例描述了图解说明上传过程的消息流图示。在这些例子中,上传在两个阶段进行。在阶段8200中,关于被上传的文件的信息被请求并且在阶段8202中,文件本身被上传。在两个阶段中,地面组件7702提示地面系统关于资源的信息。该地面系统可以是,例如在代理服务器应用程序或软件维护工具中的其他组件。

[0532] 如所描述的,机载电子分配系统7704发送消息,请求上传软件飞行器部件(消息Y1)。响应于接收到该请求,地面组件7702发送调用给地面组件接口7700,以获取特定软件飞行器部件(消息Y2)。响应该调用,如果存在软件飞行器部件,则返回软件飞行器部件的标识(消息Y3)。

[0533] 在这些例子中,如果部件不存在,则返回空值。响应于接收该消息,地面组件7702将该消息转发给机载电子分配系统7704(消息Y4)。

[0534] 在阶段8202中,机载电子分配系统7704请求软件飞行器部件(消息Y5)。响应于接收到该请求,地面组件7702从地面组件接口7700请求软件飞行器部件(消息Y6)。如果资源可用的话,地面组件接口7700返回资源(消息Y7)。如果资源不可用,则返回空值。地面组件7702接着发送软件飞行器部件给机载电子分配系统7704(消息Y8)。如果软件飞行器部件不可用,则返回错误。

[0535] 现在转向图83,根据本发明有利实施例描述了图解说明上传过程中的消息流的图示。在本例子中,数据流中存在两个阶段,阶段8300和阶段8302。在阶段8300中,请求软件飞行器部件的部分上传并且在阶段8302中进行部分软件飞行器部件的上传。如果软件飞行器部件的之前传输被中断,则进行软件飞行器部件的部分上传。

[0536] 在阶段8300中,机载电子分配系统7704发送上传请求给地面组件7702。在本例子中,该请求以部件应该从此上传的偏移或开始位置识别软件飞行器部件(消息Z1)。响应接收到此请求,地面组件7702请求软件飞行器部件(消息Z2)。

[0537] 如果存在部件,则地面组件接口7700返回软件飞行器部件。否则的话,返回空值(消息Z3)。响应接收到软件飞行器部件,地面组件7702返回响应,指出在特定偏移或开始位置可获得软件飞行器部件(消息Z4)。

[0538] 接着,在阶段8302中,机载电子分配系统7704请求在开始位置或偏移位置的软件飞行器部件(消息Z5)。地面组件7704响应接收到该请求而请求资源(消息Z6)。

[0539] 响应接收到请求,地面组件接口7700返回软件飞行器部件或者如果地面组件7702不可使用该部件则返回空值(消息Z7)。响应接收到该响应,地面组件7704在所标识的开始点或偏移开始上传软件飞行器部件(消息Z8)。如果部件不可用,则返回错误给机载电子分配系统7704。

[0540] 现在转向图84,根据本发明有利实施例描述了上传数据的过程流程图。图84中图解说明的流程可以在机载电子分配系统中实现,诸如图76中的机载电子分配系统7600。在本例子中,上行链路数据是针对软件飞行器部件。

[0541] 该流程开始于接收上行链路命令以上传软件飞行器部件(操作8400)。判断软件飞行器部件是否已经部分地上传(操作8402)。如果软件飞行器部件还没有部分地上传,则请求接收软件飞行器部件(操作8404)。该流程接着为软件飞行器部件接收数据(操作8406)。

[0542] 判断数据传输是否停止(操作8408)。可以因为许多理由停止传输。例如,软件飞行器部件的传输已经完成。在另一个例子中,在没有完成软件飞行器部件的传输的情况下可以发生中断。

[0543] 也可以因为各种事件发生中断。在一个事件中,机载电子分配系统和地面组件之间的通信链接非预期地终止。在另一个例子中,事件可以是操作员终止从软件维护工具传输软件飞行器部件。

[0544] 如果数据传输没有停止,则该流程返回至操作8406。否则,判断软件飞行器部件是否完整(操作8410)。如果软件飞行器部件是完整的,则该软件飞行器部件被存储在飞行器数据处理系统中的存储装置中(操作8412)。在本例子中,存储装置可以是图76中的大容量存储器7681。

[0545] 该流程接着返回状态给地面组件(操作8414),此后该流程结束。在本例子中,状态指出软件飞行器部件已经被完全地接收。

[0546] 再次参考操作8410,如果软件飞行器部件没有被完全接收,则软件飞行器部件已经接收的部分被存储在存储装置中(操作8416)。该流程接着存储该状态(操作8418),此后该流程结束。在该说明性例子中,状态可以标识软件飞行器部件和实际上已经接收的软件飞行器部件部分。该信息可以被用于在以后的时间重新传输软件飞行器部件的剩余部分。

[0547] 再次参考操作8402,如果软件飞行器部件已经被部分地上传,则该流程请求软件飞行器部件的未发送部分(操作8420)。该流程接着进行到操作8406以从软件飞行器部件接收数据。在操作8420中,请求可以包括尚没有被接收的软件飞行器部件数据的偏移或开始点的标识。

[0548] 现在转向图85,根据本发明有利实施例描述了下载数据的过程流程图。在图85中说明的流程可以在机载电子分配系统中实现,诸如图76中的机载电子分配系统7600。

[0549] 该流程开始于发送请求以发送下行链路文件(操作8500)。判断是否收到发送下行链路文件的数据的指示(操作8502)。如果收到发送数据的指示,则该流程发送下行链路文件的下行链路数据(操作8504)。

[0550] 接着,判断是否停止传输下行链路数据(操作1006)。可以是因为所有的数据已经被发送而停止传输。在其他情况下,例如,由于丢失通信链接或被飞行器上操作员中断而停

止传输。如果数据的传输没有停止,则该流程返回到操作8504以继续发送下行链路数据。

[0551] 如果传输已经停止,则判断是否所有的下行链路数据已经从下行链路文件发送(操作8508)。如果所有的下行链路数据已经被发送,则该流程发送完成的状态(操作8510),此后该流程结束。

[0552] 现在再次参考操作8508,如果没有发送所有的下行链路数据,则存储下行链路数据的传输状态(操作8512)。在这些例子中,状态可以被存储为图76中的状态信息7618。在本例子中,该状态可以标识下行链路文件和已发送的数据量。

[0553] 该流程也可用于发送部分下行链路文件,在该部分下行链路文件中,一部分下行链路文件已经被发送。使用这种类型的下传方式,操作8500发送请求来下传部分下行链路文件而不是整个文件。对于部分下行链路文件,如果地面组件从之前的传输发现部分下传的数据,则操作8502是主动指示。本指示也包括发送剩余部分下行链路文件的偏移或开始点。

[0554] 打包工具

[0555] 可以从各种源接收软件飞行器部件。根据特殊的实现,可以从飞行器制造商或某个第三方源接收软件飞行器部件。此外,航空公司也可以创建在其本公司的飞行器中使用的软件飞行器部件。在不同的有利实施例中,使用包来分配这些部件。

[0556] 不同的有利实施例提供计算机实现的方法、设备和计算机程序产品,其促进接收和分配处理数字化内容、计算机程序或计算机敏感的数字形式的数据的自动化。一个有利实施例包括将实际的运输包和物理介质替换为便于自动化的计算机敏感的包。另一个有利实施例是将一个或更多数字签名应用到包内的对象和包本身,从而结合功能性私有密钥基础设施提供对发送者的鉴定,完整性的认可和保证。

[0557] 在另一个有利实施例中,一种方法被用于自动化处理软件飞行器部件。从航空公司的部件管理系统外的源接收包含签名的软件飞行器部件的输入包,该输入包例如可能是电子zip文件。为该输入包和软件飞行器部件验证一组签名。响应该组有效签名,输入包被解包。在用户的自行处理下显示该输入包的内容。响应将解包的软件飞行器部件上载到软件飞行器部件管理系统或者设备中的库的请求,用认可签名将该解包的软件飞行器部件再次签名以形成已签名的认可的软件飞行器部件。一个有利实施例是,该第二个认可数字签名也作为从部件的供应商到部件的接收者的传输委托并且提供事务处理完成的认可。

[0558] 在一个有利实施例中,已签名的认可的软件飞行器部件的接收可以被置于另一包中。该包包含已签名的、接收的认可软件飞行器部件,包被签名以形成已签名包,其中该已签名的、认可的软件飞行器部件和已签名包的签名不同于在输入包中的签名组。已签名的包可以被发送到软件飞行器部件管理系统或设备中的接收库。

[0559] 在另一个有利实施例中,计算机实现的方法被用于处理额外的配置项。包含配置项的包被接收以形成接收的包。判断该包的一组签名和配置项是否有效。响应于判断该组签名为有效,则存储该配置项。

[0560] 现在转向图86,根据本发明有利实施例描述了图解说明打包工具的图示。打包工具8600被用于接收和管理包以供在环境中使用,诸如在图3中的软件飞行器部件管理设备300中。

[0561] 另外,打包工具8600可以在其他组件中实现,用于在图3的软件飞行器部件管理设

备300中创建包。例如,打包工具8600的功能可以在软件维护工具中实现,诸如图54中的软件维护工具5400。作为另一个例子,这些功能也可以在图76中的飞行器网络7601中实现,从而将诸如包中的下行链路文件的信息发送回地面网络。

[0562] 在本例子中,打包工具8600可以接收存储或包装在包8604中的软件飞行器部件8602。尽管这些例子将软件飞行器部件8602说明为包8604的内容,但在这些例子中,任何配置项可以被置入包8604中以便在图3中的软件飞行器部件管理设备300中使用。例如,配置项也可以采用文档、配置信息或其他合适信息的形式。

[0563] 打包工具8600处理包8604,用于上载到库8606。库8606可以使用图17中的库1700实现。该处理包括各种功能,诸如检查完整性和包8604中的一组签名。签名的检查可以包括对包8604和软件飞行器部件8602的签名的检查。另外,软件飞行器部件8602可以从包8604中被移除并且被检验。打包工具8600也可以将软件飞行器部件8602重新包装成另一个包,用于上载给库8606。

[0564] 现在转向图87,根据本发明有利实施例描述了图解说明打包工具的图示。打包工具8700是图86中打包工具8600的更详细说明。打包工具8700包括用户界面8702、签名8704、解包和检验8706、包8708和上载8710。用户界面8702为用户提供用户界面以操作打包工具8700。打包工具8700可以在数据处理系统中实现,诸如图2中的数据处理系统200。

[0565] 签名8704,在这些例子中提供多个不同的功能。例如,签名8704可以检查包的完整性及其配置项。该完整性可以通过检查该包的数字签名及其内容来进行。在这些例子中,签名位于可扩展标记语言文档中,该文档和被签名的内容分离。在其他实施例中,签名可以集成到被签名的配置项中。

[0566] 签名8704可以签名存在的软件飞行器部件和其他文档、文件和其他合适的的数据。解包和检验8706允许用户从包中移除软件飞行器部件和其他信息并且检验或浏览这些组件。在将包解包时,解包和检验8706从包中解压缩或移除软件飞行器部件并且将其置于所选的文件系统中。

[0567] 此外,如果在包中存在打包滑条,则该打包滑条也可以被显示。本功能的检验部分可以被用于允许用户检验存储在文件系统8712中的包8714的内容和签名有效性。包8708允许用户创建新的包并且操纵已经存在的包。

[0568] 例如,在操纵包时,用户可以组织包、添加或减去其内容。根据特殊的实现,包可以以多个不同的方式组织。例如,目录可以存储包含特定类型飞行器的软件飞行器部件的包。同样地,包可以基于其源被存储。在这些例子中,上载8710提供从打包工具8700发送包中已签名的配置项目给库的功能,该库诸如是图17的库1700。

[0569] 现在转向图88,根据本发明有利实施例描述了图解说明包的处理的消息流图示。在图88中的消息流说明了被用于处理包以上传给库的消息流。

[0570] 在本例子中,包含在处理包的过程中的不同组件包含用户8800、打包工具8802和库8804。在本例子中,当用户处理或接收输入包8806时该消息流开始。在本例子中,用户可以从不同的源接收输入包8806。例如,输入包8806可以通过因特网连接或通过某物理介质被接收,该物理介质例如为闪存或光盘。

[0571] 用户使用包检验工具打开包(操作L1)。响应该用户输入,打包工具8802将包信息显示给用户(操作L2)。用户接着检验包的内容并且选择解包该包(操作L3)。响应接收该用

户输入,打包工具8802验证该签名信息并且将包的内容解包到文件系统中(操作L4)。在输入包8806中的签名是由输入包8806中的软件飞行器部件的源生成的签名。

[0572] 然后,由用户8800生成用户输入,从而使用库上载工具将解包的软件飞行器部件上载给库(操作L5)。用户键入用户输入从而添加从解包的包位置上载的部件(操作L6)。用户接着按下上载给库按钮(操作L7)。

[0573] 响应该用户输入,打包工具8802提示库的用户8800输入登陆凭证(操作L8)。响应该提示,用户8800键入库凭证(操作L9)。打包工具8802接着提示用户输入签名密码以签名该软件飞行器部件(操作L10)。响应接收到该提示,用户8800键入密码(操作L11)。在这些例子中,该签名密码被用于创建签名,该签名被应用于软件飞行器部件的各种文件。响应从用户接收密码,打包工具8802将签名应用到不同的软件飞行器部件文件(操作L12)。

[0574] 作为该签名过程的一部分,新包被创建并且软件飞行器部件文件被置于新包中。借助这种类型的实现,在这个阶段,包中的软件飞行器部件上的数字签名与来自输入包8806的签名不同。现在应用的签名是特定用户的签名,该特定用户诸如是特定的航空公司或维护机构。

[0575] 在应用签名后,由打包工具8802启动上载部件给库8804(操作L13)。库8804上载包中的软件飞行器部件并且校验内容(操作L14)。然后,从库8804返回操作状态给打包工具8802(操作L15)。打包工具8802发送事件日志给库8804(操作L16)。事件日志由库8804上载(操作L17)。

[0576] 接着关于上载的操作状态从库8804返回给打包工具8802(操作L18)。该操作状态接着由打包工具8802提供给用户8800(操作L19)。

[0577] 现在转向图89,根据本发明有利实施例描述了图解说明打包工具的用户界面的一个实现的图示。在本例子中,用户界面8900图示说明用于实现图87中的打包工具8700的用户界面8702的组件。在本例子中,用户界面8900包括工作包列表视图8902和工作包细节视图8904。

[0578] 工作包列表视图8902显示一系列不同的包。用户可以从该视图启动项目操作8906、工作包操作8908或退出应用程序8910。项目操作8906可以被用于创建新的项目、打开存在的项目、关闭当前项目或保存当前项目。在这些例子中,工作包操作8908允许用户创建包、删除包或复制包。退出应用程序8910允许用户退出打包工具。

[0579] 另外,用户可以从工作包列表视图8902启动打开或关闭工作包8912。如果工作包是打开的,则使用工作包细节视图8904。工作包细节视图8904提供用户界面,该界面根据正在被处理的包的特定类型显示不同的功能。

[0580] 另外,用户可以从工作包列表视图8902和从工作包细节视图8904访问工具8914。工具8914提供各种功能,诸如检查包的完整性、解包和检验包以及检查兼容性和设置优先级。在本例子中,工具8914提供功能8916、8918、8920、8922和8924。功能8916显示关于数字签名和配置项的签名状态的信息。签名状态的例子是生产、批准和来源。

[0581] 功能8918将签名的部件和/或包中的资源解包并且将这些组件置于文件系统中。功能8920提供对包内容的检验、验证包和组件签名并且管理包文件。功能8922允许用户检查软件飞行器部件与飞机的机载数据加载功能(ODLF)的兼容性。功能8924允许用户编辑各种属性和偏好。所描写的功能作为功能的说明性例子提供,这些功能可以在工具8914中提

供。当然,其他的功能可以被用于补充或替换所描述的功能。

[0582] 现在参考图90,根据本发明有利实施例描述了图解说明检验和解包包时的数据流图示。在图90中说明的数据流可以在图87的打包工具8700中的解包和检验8706中实现。

[0583] 在本例子中,给出两个对话包或视图,检验和解包视图9000和包检验视图9002。检验和解包视图9000显示给用户并且允许用户进行关于打包工具已经接收的包的不同操作。例如,用户可以选择一个操作来操控包。该操作可以是,例如删除或转移一组包文件。

[0584] 如果用户从检验和解包视图9000中选择该操作,所选的包文件就被转移或删除(操作9004)。然后,包列表被刷新(操作9006)并且该流程返回到检验和解包视图9000。

[0585] 如果用户选择的操作诸如为移除一个位置,则该流程接着修改位置列表(操作9008)。该位置的优选接着被存储(操作9010),该流程接着返回到如上所述的操作9006。这个位置优选是用户选择的路径或目录。在该方法中,用户可以从一组目录或在其中存储包的一组位置中删除位置。

[0586] 在检验和解包视图9000处,如果用户在视图中的新位置处键入用户输入,该用户被提示输入新位置(操作9011)。一旦用户键入新位置信息,该流程进入到如上所述的操作9008。如果用户选择或决定检验包,则该流程转移到包检验视图9002。在该用户界面中,用户可以进行关于包的各种操作。例如,用户可以选择更新包信息。然后,从文件读取包信息(操作9014)。该流程接着更新对话包控件的数据,从而将包信息显示给用户(操作9016)。

[0587] 当包检验视图9002被显示时,用户可以选择另一个操作,诸如将包解包。本操作的选择导致包签名被验证(操作9018)。如果签名是有效的,则包被解包(操作9020)。该流程给出解包该包的结果,并伴随着在包检验视图9002中显示任何打包滑条内容(操作9022)。然后该流程返回到检验和解包视图9000。

[0588] 在操作9018中,如果包的签名是无效的,则该流程给出验证结果(操作9024)。如果验证对于包签名来说是错误的,则这些在包检验视图9002中给出的结果可以包括签名问题是致命的指示。

[0589] 在包检验视图9002中,如果用户选择验证包,则该流程验证包签名(操作9026)。如果包签名是有效的,则接着验证每个配置项签名(操作9028)。在操作9026和9028中,该流程进行到操作9024以显示验证的结果。如果配置项签名是无效的,则不同于在包签名是无效的时候发生致命问题,而是给出警告。在包检验视图9002中,如果用户关闭对话包,则该流程返回到检验和解包视图9000。

[0590] 现在转向图91,根据本发明有利实施例描述了图解说明在创建包的过程中的数据流图。在图91中说明的流程可以在打包工具中实现,诸如图87中的打包工具8700。更具体地,显示在图91中的不同操作可以在图87的包8708中实现。

[0591] 在本例子中,该流程开始于打开新的或已有项目(操作9100)。然后,该流程创建工作包(操作9102)。在创建原始的已签名配置项时,该流程始于接收原始包元数据和配置项标识符(操作9104)。在这些例子中,配置项是由一组存储在包中的文件组成的单个项。每个配置项具有唯一的标识符。配置项可以是,例如,软件飞行器部件、相关文档或某个其他文件。

[0592] 用户接着定位到文件系统上的配置项的数据目录并且为这些选择的文件键入元数据(操作9106)。选择数据文件的目录,因为特定配置项可以包括多于一个文件。例如,软

件飞行器部件可以包括可执行文件、配置文件和动态链接库。

[0593] 接着,该流程验证用户键入的元数据条目(操作9108)。在操作9108中,该流程可以判断元数据条目是否符合一组规则。这些规则可要求包含特定数量信息和特定类型信息的特定类型的配置项。例如,对于软件飞行器部件,软件飞行器部件的源或制造商以及飞行器类型的标识可以输入为元数据。此外,该元数据可以标识接收软件飞行器部件的特定飞行器。

[0594] 该流程可以根据工作包的类型验证配置项(操作9110)。接着,该流程接收用户密码(操作9112)。该流程接着为配置项创建数字签名的可扩展标记语言文件并且将数字签名的可扩展标记语言文件与配置项一起存储在文件系统中(操作9114)。该流程继续到保存项目(操作9116)。用户可以在任何这些不同操作的过程中选择停止和保存项目并且在其他时间继续该项目。

[0595] 用户接着定位到文件系统中的资产的数据目录并且为这些所选的文件键入元数据(操作9118)。然后,该流程验证由用户键入的元数据条目(操作9120)。该流程接收用户密码(9122)。该流程接着为资产创建数字签名的可扩展标记语言文件并且将其存储在资产的目录中(操作9124)。

[0596] 现在转向图92,根据本发明有利实施例描述了处理所接收的包的过程流程图。在图92中说明的流程可以在软件组件中实现,诸如图87中的打包工具8700。更具体地,该流程可以在图87的解包和检验8706中实现。

[0597] 该流程开始于接收包(操作9200)。在本例子中,包可以通过各种源被接收。例如,物理介质可以被连接到或置于在其中执行该流程的数据处理系统中。在其他实施例中,包可以通过通信链接诸如网络链接接收。

[0598] 该流程给出关于包的信息(操作9202)。在本操作中,信息可以通过图形用户界面给出。该信息可以包括,例如,包的制造货源、包中内容的标识、包的大小和其他合适的信息。然后,判断是否解包包(操作9204)。该判断可以通过接收用户输入做出。

[0599] 如果包要被解包,则该流程验证包的签名(操作9206)。在这些例子中,签名可以使用私钥进行签名。位于包中的公钥可以被用于判断名单和文件摘要是否有效。本验证也可以被用于判断包实际上是否已经由源创造并且保持为未修改或已经被篡改。

[0600] 判断包的签名是否有效(操作9208)。如果包的签名有效,则该流程将包含软件飞行器部件的包解包并且将其置于文件系统上(操作9210)。配置项签名在解包包时不必是有效的。在探测到一个或多个无效的配置项签名时由用户决定是否继续解包包。在这些例子中,如果包签名是有效的,则软件飞行器部件被解包并且存储在文件系统中,如操作9210中描述的。此后该流程结束。

[0601] 再次参看操作9208,如果包签名不是有效的,则返回错误(操作9212),此后该流程结束。参考操作9204,如果用户决定不解包包,则该流程同样也结束。

[0602] 该工具支持 workflow 状态和动态。如在上面图2中所讨论的用户界面的实现允许用户创建、验证和完成包。包始于草稿状态,包的每个组件(诸如部件和相关的文档)同样也是这样。随着本流程的进行过,包以及每个组件的状态从草稿变为在完成中,并且在包被签名后变为完成的。该实现也允许用户添加、删除或修改包的任何组件。此外,任何添加、删除或修改将会导致相关组件和包的当前状态的改变,并且因此要求重新验证和重新签名。该状

态在包列表视图和包细节视图中都以图形指示。在提供灵活性和支持工作流的同时,本功能进一步确保完成的包的完整性。

[0603] 该工具支持动态发布/分配工作流状态。如上面在图4中讨论的用户界面的实现允许用户创建、验证和完成包。包开始于草稿状态,包的每个组件(诸如部件和相关的文档)同样也是这样。在本流程进行过程中,包以及每个组件的状态从草稿变为在完成中,并且在包被签名后改变为完成的。该实现也允许用户添加、删除或修改包的任何组件。此外,任何添加、删除或修改将会导致相关组件和包的当前状态的改变,并且因此要求重新验证和重新签名。该状态在包列表视图和包细节视图中都以图形指示。在提供灵活性和支持工作流的同时,本功能进一步确保完成的包的完整性。

[0604] 在不同描述的实施例中的流程图和框图可以说明设备、方法和计算机程序产品的一个或多个可能实现的体系结构、功能和操作。就这一点而言,流程图或框图的每个块可以表示模块、程序段或部分计算机可用或可读的程序代码,其包括用于实现指定的一个或多个功能的一个或更多可执行指令。在某些替代的实现中,在块中标注的一个或多个功能可以不以图中标注的顺序发生。例如,在某些情况下,依据所包含的功能,相继显示的两个块可以基本同时执行,或者有时块可以以相反的顺序执行。

[0605] 另外,提供在本公开中说明的软件组件、硬件组件和数据结构的不同框图,目的是描述不同的有利实施例可以被实现的方式,并且并不意味着不同实施例可能采取的形式。例如,一些框图说明可以在软件实现中组合或分割的功能块。同样,在这些例子中说明的硬件和体系结构可以在不同的有利实施例中不同。同样,给出不同的图形用户界面的例子,目的是说明一种实现用户界面的方式。这些例子也不意味着限制实现不同有利实施例的方式。

[0606] 不同有利实施例可以采用完全基于硬件的实施例、完全基于软件的实施例或包含硬件和软件元素两者的实施例的形式。一些实施例在软件中实现,该软件包括但不限于例如固件、常驻软件和微代码的形式。

[0607] 此外,不同的实施例可以采用计算机程序产品的形式,该计算机程序产品可以从计算机可用或计算机可读的介质访问,所述介质提供供计算机或执行指令的任何装置或系统使用或结合其使用的程序代码。出于本公开的目的,计算机可用或计算机可读的介质通常可以是有形的设备,该有形的设备可以包含、存储、通信、传播或传输指令执行系统、设备或装置使用的或结合执行系统、设备或装置使用的程序。

[0608] 计算机可用或计算机可读介质可以是,例如而限于电子、磁、光、电磁、红外或半导体系统,或传播介质。计算机可读介质的非限制性的例子包括半导体或固态存储器、磁带或移动式计算机磁盘、随机存取存储器(RAM)、只读存储器(ROM)、硬磁盘、软磁盘和光盘。光盘可以包括只读存储光盘(CD-ROM)、读写光盘(CD-R/W)和DVD。

[0609] 此外,计算机可用或计算机可读介质可以包含或存储计算机可读或可用程序代码,从而在计算机上执行计算机可读或可用程序代码时,计算机可读或可用程序代码的执行使得计算机在一个或更多通信链接上传输另一计算机可读或可用程序代码。每个通信链接可以是有线的或无线的。

[0610] 适合于存储和/或执行计算机可读或计算机可用程序代码的数据处理系统将包括通过诸如系统总线的通信组织直接或间接地耦合于存储器元件的一个或更多处理器。存储

器元件可以包括在实际执行程序代码期间使用的本地存储器、大容量存储器和高速缓冲存储器,该高速缓冲存储器提供至少一些计算机可读或计算机可用程序代码的临时存储,从而降低在执行代码期间从大容量存储器检索代码的时间量。

[0611] 输入/输出(或I/O)装置可以直接地或通过中间I/O控制器耦合到系统。这些装置可以包括,例如而不限于键盘、触摸屏显示器和指向装置。不同的通信适配器可以被耦合到系统,从而确保数据处理系统通过中介私人或公共网络耦合到其他数据处理系统或远程打印机或存储装置。非限制性的例子是,调制解调器和网络适配器,它们只是当前可用类型的通信适配器重的几种。

[0612] 用于分配可加载软件飞行器部件(LSAP)分配的方法和设备

[0613] 实施例可以要求权利如下:

[0614] A1.一种软件飞行器部件管理设备,包括:与数据处理系统关联的库,该库被配置为管理一组飞行器的一组软件飞行器部件;能够接收多个软件飞行器部件并且发送该组软件飞行器部件给库的接收模块;与库的通信中的代理服务器,其中该代理服务器能够服务一组飞行器客户系统;该组飞行器中的一飞行器的机载电子分配系统,其中机载电子分配系统是该组飞行器客户系统中的一飞行器客户系统并且能够通过代理服务器从库为该飞行器接收软件飞行器部件;以及便携式软件维护工具,其中,该便携式软件维护工具提供发送软件飞行器部件给该飞行器的机载电子分配系统的替代途径。

[0615] A2.权利要求A1的软件飞行器部件管理设备,其中,库被配置为创建包含软件飞行器部件的包,其中所述软件飞行器部件在包中被发送给机载电子分配系统。

[0616] A3.权利要求A1的软件飞行器部件管理设备,其中,所述代理服务器与数据处理系统相关联。

[0617] A4.权利要求A1的软件飞行器部件管理设备,其中所述便携式软件维护工具连接到所述代理服务器以接收软件飞行器部件,该便携式软件维护工具被转移到包含该机载电子分配系统的飞行器上,并且连接到该机载电子分配系统以传输软件飞行器部件给机载电子分配系统。

[0618] A5.权利要求A1的软件飞行器部件管理设备,其中,所述机载电子分配系统从飞行器上的一组外场可更换单元上传数据到代理服务器以形成下行链路数据,并且其中所述代理服务器发送下行链路数据给库,用于存储。

[0619] A6.权利要求A1的软件飞行器部件管理设备,其中,所述软件飞行器部件在包中从库转移到机载电子分配系统。

[0620] A7.用于管理多个软件飞行器部件的计算机实现的方法,所述计算机实现的方法包括:在库中存储软件飞行器部件;为软件飞行器部件识别飞行器以形成已识别的飞行器;从库发送软件飞行器部件给代理服务器;以及从代理服务器发送软件飞行器部件给已识别的飞行器上的机载电子分配系统。

[0621] A8.权利要求A7的计算机实现的方法,其中从代理服务器发送软件飞行器部件给已识别的飞行器上的机载电子分配系统的步骤包括:使用通信链接从代理服务器发送软件飞行器部件给已识别的飞行器上的机载电子分配系统。

[0622] A9.权利要求A7的计算机实现的方法,其中从代理服务器发送软件飞行器部件给已识别的飞行器上的机载电子分配系统的步骤包括:使用便携式软件维护工具从代理服务

器发送软件飞行器部件给已识别的飞行器上的机载电子分配系统。

[0623] A10. 权利要求A7的计算机实现的方法,其中存储步骤包括:从供应商接收软件飞行器部件以形成已接收的软件飞行器部件,处理软件飞行器部件以形成已处理的软件飞行器部件,并且在库中存储已处理的软件飞行器部件。

[0624] A11. 权利要求A7的计算机实现的方法进一步包括:通过代理服务器从库发送命令给机载电子分配系统以从飞行器下载数据;以及响应于发送命令而通过代理服务器接收库中的数据。

[0625] A12. 权利要求A7的计算机实现的方法进一步包括:从库发送软件飞行器部件给便携式计算机上的软件维护工具;将具有软件飞行器部件的便携式计算机转移到飞行器上;在将便携式计算机转移到飞行器后建立软件维护工具和飞行器上的机载电子分配系统之间的通信链接以形成建立的通信链接;以及通过建立的通信链接从软件维护工具发送软件飞行器部件给机载电子分配系统。

[0626] 用于管理软件飞行器部件的计算机实现的方法和设备。软件飞行器部件管理设备包括库、接收模块、代理服务器、飞行器上的机载电子分配系统和便携式软件维护工具。地面数据处理系统上的库为飞行器存储和管理软件飞行器部件。接收模块能够从供应商接收软件飞行器部件并且将所述软件飞行器部件发送给库。代理服务器与库通信并且能够将软件分配给多个飞行器客户系统。机载电子分配系统是飞行器客户系统并且能够通过代理服务器从库为飞行器接收软件飞行器部件。便携式软件维护工具提供将软件飞行器部件发送给飞行器的机载电子分配系统的替代途径。

[0627] 打包工具

[0628] 实施例可以要求权利如下:

[0629] B1. 用于处理软件飞行器部件的计算机实现的方法,该计算机实现的方法包括:从软件飞行器部件管理设备外的源处接收包含软件飞行器部件的包;为输入包和软件飞行器部件验证一组签名;响应被验证的该组签名,将输入包解包;显示输入包的内容;解包所述包以获取软件飞行器部件;响应将软件飞行器部件上载到软件飞行器部件管理设备中的库中的请求,判断软件飞行器部件是否符合规定;响应于软件飞行器部件符合规定,将软件飞行器部件签名以形成已签名的软件飞行器部件;将已签名的软件飞行器部件置于包内,为包含已签名的软件飞行器部件的包签名以形成已签名包,其中已签名的软件飞行器部件和已签名包的签名与输入包中的该组签名不同;以及发送已签名的包给软件飞行器部件管理设备中的库。

[0630] B2. 权利要求B1的计算机实现的方法进一步包括:识别软件飞行器部件的元数据和其他配置项。

[0631] B3. 权利要求B1的计算机实现的方法,其中,所述软件飞行器部件是第一个签名的软件飞行器部件并且已签名的软件飞行器部件是使用和第一个签名的软件飞行器部件不同的签名被签名。

[0632] B4. 权利要求B1的计算机实现的方法,其中,已签名的软件飞行器部件或包的签名来自于制造商、航空公司或供应商。

[0633] B5. 权利要求B1的计算机实现的方法,其中,存储步骤包括:将软件飞行器部件存储到文件系统中的目录下。

[0634] B6. 权利要求B1的计算机实现的方法进一步包括:显示用户界面,该用户界面能够接收用户的输入从而在多个所存储的软件飞行器部件上进行操作。

[0635] B7. 用于处理配置项的计算机实现的方法,该计算机实现的方法包括:接收包含配置项的包以形成接收的包;判断包和配置项的一组签名是否有效;以及响应该组签名是有效的判断保存配置项。

[0636] B8. 权利要求B7的计算机实现的方法进一步包括:判断配置项是否有效。

[0637] B9. 权利要求B7的计算机实现的方法,其中,判断配置项是否有效的步骤包括:比较配置项与规定。

[0638] B10. 权利要求B9的计算机实现的方法,其中所述配置项是软件飞行器部件并且所述规定是软件飞行器部件标准。

[0639] B11. 权利要求B7的计算机实现的方法进一步包括:响应于上载有效的配置项的用户输入,将有效的配置项签名;将有效的配置项置于包中;签名包;以及发送包给软件飞行器部件管理设备中的库。

[0640] B12. 权利要求B7的计算机实现的方法进一步包括:响应于用户输入,组织多个配置项,所述配置项包括文件系统中的配置项。

[0641] B13. 权利要求B7的计算机实现的方法进一步包括:显示能够接收用户输入的用户界面,从而操纵一组配置项。

[0642] B14. 权利要求B7的计算机实现的方法进一步包括:显示能够支持和指出一组配置项的动态发布/分配状态的用户界面。

[0643] B15. 权利要求B7的计算机实现的方法,其中,该组签名包括针对配置项的第一个签名和针对包的第二个签名。

[0644] B16. 权利要求B7的计算机实现的方法,其中,所述包包括主要的配置项和一组额外的配置项。

[0645] B17. 一种数据处理系统,包括:总线;连接到总线的存储装置,其中该存储装置包括程序代码;连接到总线的处理器单元,其中处理器单元执行所述程序代码以从软件飞行器部件管理设备外部的源接收包含软件飞行器部件的输入包;验证输入包和软件飞行器部件的一组签名;响应该组签名是有效的,解包输入包;显示输入包的内容;存储软件飞行器部件;响应将软件飞行器部件上载至软件飞行器部件管理设备中的库的请求,判断软件飞行器部件是否符合规定;响应于软件飞行器部件符合规定,将软件飞行器部件签名以形成已签名的软件飞行器部件;将已签名的软件飞行器部件置于包内;为包含已签名的软件飞行器部件的包签名以形成已签名的包,其中已签名的软件飞行器部件和已签名的包的签名与输入包中的该组签名不同;以及发送已签名的包给软件飞行器部件管理设备中的库。

[0646] B18. 权利要求17的数据处理系统,其中所述处理器单元进一步执行程序代码以识别软件飞行器部件的元数据。

[0647] B19. 权利要求17的数据处理系统,其中所述规定是软件飞行器部件的工业标准。

[0648] B20. 权利要求17的数据处理系统,其中已签名的软件飞行器部件和已签名包的签名来自于航空公司、供应商或制造商。

[0649] B21. 权利要求17的数据处理系统,其中在执行存储软件飞行器部件的程序代码的过程中,处理器单元执行程序代码以将软件飞行器部件存储在文件系统中的目录下。

[0650] 计算机实现的用于处理软件飞行器部件的方法、设备和计算机程序产品。从源接收包含软件飞行器部件的包。验证包和软件飞行器部件的签名。如果签名有效,则解包该包。显示输入的包的内容。响应将存储的部件上传至软件飞行器部件管理设备中的库的请求,判断所存储的部件是否符合规定。如果所存储的部件符合规定,则该部件被签名。已签名的部件被置于包中并且被签名以形成已签名的包,其中已签名的软件飞行器部件和已签名的包的签名与输入包中的该组签名不同。已签名的包被发送给库。

[0651] 用于分配软件飞行器部件的代理服务器

[0652] 实施例可以要求权利如下:

[0653] C1. 用于管理软件飞行器部件的计算机实现的方法,该计算机实现的方法包括:从飞行器网络数据处理系统中的库检索一组软件飞行器部件和关于该组软件飞行器部件的元数据以形成一组检索到的软件飞行器部件;将该组软件飞行器部件存储在文件系统中;将元数据存储在数据库中;发送该组软件飞行器部件和元数据给机载电子分配系统、地面组件接口和机载电子分配系统之间的通信链接;以及保持关于通信链接的活动性的状态信息。

[0654] C2. 权利要求C1的计算机实现的方法进一步包括:发送状态信息给库。

[0655] C3. 权利要求C2的计算机实现的方法,其中所述状态信息包括关于该组软件飞行器中的一软件飞行器部件是否准备好安装在飞行器上的外场可更换单元上的标识。

[0656] C4. 权利要求C1的计算机实现的方法,其中该组软件飞行器部件是第一组软件飞行器部件,该元数据是第一元数据,该机载电子分配系统是第一机载电子分配系统以及该通信链接是第一通信链接并且进一步包括:发送该组软件飞行器部件和该元数据给该机载电子分配系统,同时通过地面组件接口和第二机载电子分配系统之间的第二通信链接发送第二组软件飞行器部件和第二元数据给第二机载电子分配系统。

[0657] C5. 权利要求C1的计算机实现的方法进一步包括:从库接收一组命令以形成接收的命令组;通过地面组件接口和第二机载电子分配系统之间的第二通信链接发送接收的命令组给第二机载电子分配系统。

[0658] C6. 权利要求C5的计算机实现的方法,其中该组命令包括删除命令和下行链路命令中的至少一个。

[0659] C7. 权利要求C5的计算机实现的方法进一步包括:通过地面组件接口和第二机载电子分配系统之间的第二通信链接接收下行链路数据;在文件系统中存储下行链路数据以形成存储的下行链路数据;以及发送存储的下行链路数据给库。

[0660] C8. 一种设备,包括:文件系统;数据库;存储在文件系统中的一组软件飞行器部件;存储在是数据库中的一组命令;能够与多个飞行器上的多个机载电子分配系统交换信息的地面组件;控制进程,其中控制进程能够从库接收该组命令和该组软件飞行器部件并且发送该组命令和该组软件飞行器部件给多个飞行器;以及数据处理系统,其中文件系统、数据库、该组软件飞行器部件、该组命令、地面组件和控制进程是数据处理系统上的软件组件。

[0661] C9. 权利要求C8的设备,其中所述控制进程还能够发送该组命令和该组软件飞行器部件给便携式数据处理系统上的软件管理工具。

[0662] C10. 权利要求C8的设备,其中所述地面组件能够检索下行链路数据和将下行链路

数据存储在文件系统中的。

[0663] C11. 权利要求C8的设备,其中所述控制进程能够将下行链路数据从文件系统传输到库。

[0664] C12. 权利要求C8的设备,其中在包中从库接收该组软件飞行器部件中的软件飞行器部件。

[0665] C13. 权利要求C8的设备,其中所述设备位于机场或维护机构。

[0666] C14. 计算机程序产品,包括:计算机可读介质;存储在计算机可读介质上的程序代码,用于从飞行器网络数据处理系统中的库中检索一组软件飞行器部件和关于该组软件飞行器部件的元数据,形成一组检索到的软件飞行器部件;存储在计算机可读介质上的程序代码,用于在文件系统中存储该组软件飞行器部件;存储在计算机可读介质上的程序代码,用于在数据库中存储元数据;存储在计算机可读介质上的程序代码,用于发送该组软件飞行器部件和元数据给机载电子分配系统、地面组件接口和机载电子分配系统之间的通信链接;以及存储在计算机可读介质上的程序代码,用于保持关于通信链接的活动性的状态信息。

[0667] C15. 权利要求C14的计算机程序产品进一步包括:存储在计算机可读介质上的程序代码,用于发送状态信息给库。

[0668] C16. 权利要求C15的计算机程序产品,其中所述状态信息包括对该组软件飞行器部件中的软件飞行器部件是否准备好安装在飞行器上的外场可更换单元的标识。

[0669] C17. 权利要求C14的计算机程序产品,其中该组软件飞行器部件是第一组软件飞行器部件,该元数据是第一元数据,该机载电子分配系统是第一机载电子分配系统以及该通信链接是第一通信链接并且进一步包括:存储在计算机可读介质上的程序代码,用于发送该组软件飞行器部件和该元数据给该机载电子分配系统并同时通过地面组件接口和第二机载电子分配系统之间的第二通信链接发送第二组软件飞行器部件和第二元数据给第二机载电子分配系统。

[0670] C18. 权利要求C14的计算机程序产品进一步包括:存储在计算机可读介质上的程序代码,用于从库接收一组命令,以形成接收的命令组;存储在计算机可读介质上的程序代码,用于通过地面组件接口和第二机载电子分配系统之间的第二通信链接发送接收的命令组给第二机载电子分配系统。

[0671] C19. 权利要求C18的计算机程序产品,其中该组命令包括删除命令和下行链路命令中的至少一个。

[0672] C20. 权利要求C18的计算机程序产品进一步包括:存储在计算机可读介质上的程序代码,用于通过地面组件接口和第二机载电子分配系统之间的第二通信链接接收下行链路数据;存储在计算机可读介质上的程序代码,用于在文件系统中存储下行链路数据以形成存储的下行链路数据;以及存储在计算机可读介质上的程序代码,用于发送所存储的下行链路数据给库。

[0673] 用于管理软件飞行器部件的计算机实现的方法、设备和计算机程序产品。在一个有利实施例中,一种方法包括从库检索一组软件飞行器部件和关于该组软件飞行器部件的元数据以形成一组检索到的软件飞行器部件。将该组软件飞行器部件存储在文件系统中。将元数据存储在数据库中。发送该组软件飞行器部件和元数据给机载电子分配系统、地面

组件接口和机载电子分配系统之间的通信链接。保持关于通信链接的活动性的状态信息。

[0674] 用于处理飞行器网络中的命令的方法和设备

[0675] 实施例可以要求权利如下：

[0676] D1. 计算机实现的方法，用于处理网络数据处理系统中的命令，该计算机实现的方法包括：为多个飞行器中的飞行器创建一组命令；将该组命令存储在网络数据处理系统中的库中的一组队列以形成存储的命令组；响应于事件而发送存储的命令组给在网络数据处理系统上运行的代理服务器应用程序以形成已分配的命令组；建立代理服务器应用程序和在飞行器上的飞行器数据处理系统上运行的机载电子分配系统之间的连接；由机载电子分配系统从代理服务器应用程序请求分配的命令组；由机载电子分配系统从代理服务器应用程序接收分配的命令组以形成接收的命令组；由机载电子分配系统处理接收的命令组以形成经处理的命令组；以及如果经处理的命令组是一组上行链路命令则将一组软件飞行器部件转发给飞行器上的一组目标数据处理系统。

[0677] D2. 权利要求1的计算机实现的方法，其中处理步骤包括：识别命令组中命令的命令类型以形成已识别的命令；如果该命令是相应软件飞行器部件的上行链路命令，则请求该组软件飞行器部件中相应的软件飞行器部件，以形成已请求的软件飞行器部件；以及接收已请求的软件飞行器部件。

[0678] D3. 权利要求D1的计算机实现的方法，其中分配的命令组在一组包中被发送，其中在该组包中的每一个包都由数字签名签名。

[0679] D4. 权利要求D3的计算机实现的方法，其中处理步骤包括：验证该组包中的每一个包的数字签名。

[0680] D5. 权利要求D2的计算机实现的方法，其中处理步骤进一步包括：如果已识别的命令是下行链路命令则从机载电子分配系统请求下行链路数据。

[0681] D6. 权利要求D2的计算机实现的方法进一步包括：由机载电子分配系统从飞行器上的所选的数据处理系统检索数据以形成检索到的数据；由机载电子分配系统将检索到的数据置于包中以形成下行链路数据；通过所述连接从机载电子分配系统发送下行链路数据给代理服务器应用程序以形成接收的下行链路数据；以及发送接收的下行链路数据给库。

[0682] D7. 权利要求D2的计算机实现的方法，其中处理步骤包括：如果已识别的命令是删除命令，则识别数据处理系统以形成目标数据处理系统；以及发送已识别的命令给目标数据处理系统以删除目标数据处理系统上的资源。

[0683] D8. 用于处理命令的计算机实现的方法，该计算机实现的方法包括：识别网络数据处理系统中的库中的一组命令以形成一组已识别的命令；发送该组已识别的命令给在网络数据处理系统上运行的代理服务器应用程序以形成一组分配的命令；建立代理服务器应用程序和在飞行器上的飞行器数据处理系统上运行的机载电子分配系统之间的连接；通过连接从代理服务器应用程序发送该组分配的命令给机载电子分配系统以形成一组接收的命令；由机载电子分配系统处理该组接收的命令以形成一组经处理的命令；如果该组经处理的命令是一组上行链路命令，则使用该组经处理的命令将位于飞行器数据处理系统上的所存储的该组软件飞行器部件转发给飞行器上的一组目标数据处理系统。

[0684] D9. 权利要求D8的计算机实现的方法进一步包括：发送该组软件飞行器部件给飞行器数据处理系统以形成存储的软件飞行器部件组。

[0685] D10. 权利要求D8的计算机实现的方法,其中所述处理步骤包括:识别命令组中一命令的命令类型以形成已识别的命令;并且执行该已识别的命令。

[0686] D11. 权利要求D10的计算机实现的方法,其中已识别的命令是删除命令并且其中运行步骤包括:从该组目标数据处理系统删除对应的软件飞行器部件。

[0687] D12. 权利要求D10的计算机实现的方法,其中已识别的命令是下行链路命令并且其中运行步骤包括:通过代理服务器应用程序从该组目标数据处理系统中的所选目标数据处理系统将数据下载到库。

[0688] D13. 权利要求D8的计算机实现的方法,其中通过连接从代理服务器应用程序发送该组分配的命令给机载电子分配系统以形成该组接收的命令的步骤包括:经由连接通过在便携式数据处理系统上运行的软件管理工具从代理服务器应用程序发送该组分配的命令给机载电子分配系统以形成一组接收的命令。

[0689] D14. 计算机程序产品,包括:计算机可读介质;存储在计算机可读介质上用于为多个飞行器中的飞行器创建一组命令的程序代码;存储在计算机可读介质上用于将该组命令存储在网络数据处理系统中的库中的一组队列中以形成存储的命令组的程序代码;存储在计算机可读介质上用于响应事件而发送存储的命令组给在网络数据处理系统上运行的代理服务器应用程序以形成分配的命令组的程序代码;存储在计算机可读介质上用于建立代理服务器应用程序和行在飞行器上的飞行器数据处理系统上运行的机载电子分配系统之间的连接的程序代码;存储在计算机可读介质上用于由机载电子分配系统从代理服务器应用程序请求分配的命令组的程序代码;存储在计算机可读介质上用于由机载电子分配系统从代理服务器应用程序接收被分配的命令组以形成被接收的命令组的程序代码;存储在计算机可读介质上用于由机载电子分配系统处理接收的命令组以形成已处理的命令组的程序代码;以及存储在计算机可读介质上的程序代码,如果已处理的命令组是一组上行链路命令,则使用已处理的命令组将一组软件飞行器部件转发给飞行器上的一组目标数据处理系统。

[0690] D15. 权利要求D14的计算机程序产品,其中存储在计算机可读介质上用于由机载电子分配系统处理接收的命令组以形成已处理的命令组的程序代码包括:存储在计算机可读介质上用于识别命令组中一命令的命令类型以形成已识别的命令的程序代码;存储在计算机可读介质上的程序代码,如果命令是相应软件飞行器部件的上行链路命令则请求该组软件飞行器部件中相应的软件飞行器部件以形成已请求的软件飞行器部件;以及存储在计算机可读介质上用于接收已请求的软件飞行器部件的程序代码。

[0691] D16. 权利要求D14的计算机程序产品,其中已分配的命令组在一组包中被发送,其中该组包中的每一个包都用数字签名签名。

[0692] D17. 权利要求D16的计算机程序产品,其中,存储在计算机可读介质上用于由机载电子分配系统处理接收的命令组以形成已处理的命令组的程序代码包括:存储在计算机可读介质上用于验证该组包中的每个包的数字签名的程序代码。

[0693] D18. 权利要求D15的计算机程序产品,其中,存储在计算机可读介质上用于由机载电子分配系统处理接收的命令组以形成已处理的命令组的程序代码进一步包括:存储在计算机可读介质上的程序代码,如果已识别的命令是下行链路命令则从机载电子分配系统请求下行链路数据。

[0694] D19. 权利要求D18的计算机程序产品进一步包括:存储在计算机可读介质上用于由机载电子分配系统从飞行器上所选的数据处理系统检索数据以形成检索到的数据的程序代码;存储在计算机可读介质上用于由机载电子分配系统将检索到的数据置于包中以形成下行链路数据的程序代码;存储在计算机可读介质上用于通过连接从机载电子分配系统发送下行链路数据给代理服务器以形成接收的下行链路数据的程序代码;以及存储在计算机可读介质上用于发送接收的下行链路数据给库的程序代码。

[0695] D20. 权利要求D15的计算机程序产品进一步包括:存储在计算机可读介质上用于由机载电子分配系统处理接收的命令组以形成已处理的命令组的程序代码包括:存储在计算机可读介质上的程序代码,如果已识别的命令是删除命令则识别数据处理系统以形成目标数据处理系统;以及存储在计算机可读介质上的程序代码,用于发送已识别的命令给目标数据处理系统以删除目标数据处理系统上的资源。

[0696] 用于在网络中处理命令的方法、设备和程序代码。一处理为飞行器创建一组命令。这些命令在网络中的库的队列中。响应一事件,存储的命令组被发送给在网络数据处理系统上执行的代理服务器应用程序。在代理服务器应用程序和飞行器上的飞行器数据处理系统上的机载电子分配系统之间建立连接。由机载电子分配系统请求这些命令。由机载电子分配系统从代理服务器应用程序接收分配的命令。由机载电子分配系统处理接收的命令并且如果命令是上行链路命令,则使用处理的命令将一组软件飞行器部件转发给飞行器上的一组目标数据处理系统。

[0697] 软件维护工具

[0698] 实施例可要求权利如下:

[0699] E1. 用于管理软件飞行器部件的计算机实现的方法,该计算机实现的方法包括:通过地面网络建立便携式数据处理系统和源之间的连接以形成已建立的连接;通过已建立的连接从源检索一组上行链路命令;通过已建立的连接从源检索与该组上行链路命令相对应的一组软件飞行器部件以形成一组检索到的软件飞行器部件;在便携式数据处理系统中存储该组检索到的软件飞行器部件以形成一组存储的软件飞行器部件;断开具有该组存储的软件飞行器部件的便携式数据处理系统与地面网络的连接;将具有该组存储的软件飞行器部件的便携式数据处理系统连接到飞行器中的飞行器网络;通过便携式数据处理系统中的地面组件从该组上行链路命令发出一上行链路命令给飞行器网络;以及通过地面组件发送与该上行链路命令对应的存储的软件飞行器部件给飞行器网络。

[0700] E2. 权利要求E1的计算机实现的方法进一步包括:从飞行器网络接收一组下行链路文件以形成一组接收的下行链路文件。

[0701] E3. 权利要求E1的计算机实现的方法进一步包括:识别多个软件飞行器部件中的每个软件飞行器部件是否成功地发送给飞行器网络以形成状态;以及响应与地面网络建立的后续连接,发送该状态给源。

[0702] E4. 权利要求E1的计算机实现的方法,其中发送步骤包括:通过地面组件发送存储的软件飞行器部件给飞行器网络上的机载电子分配系统。

[0703] E5. 权利要求E1的计算机实现的方法进一步包括:通过一组图形用户界面控制便携式数据处理系统的操作。

[0704] E6. 权利要求E1的计算机实现的方法进一步包括:通过该组图形用户界面中的图

形用户界面创建上行链路命令,以发送便携式数据处理系统上存储的软件飞行器部件给飞行器网络。

[0705] E7. 权利要求E1的计算机实现的方法进一步包括:从便携式介质接收软件飞行器部件以形成接收的软件飞行器部件;以及将接收的软件飞行器部件存储在便携式数据处理系统中以形成存储的软件飞行器部件。

[0706] E8. 权利要求E1的计算机实现的方法进一步包括:通过图形用户界面创建上行链路命令以发送存储的软件飞行器部件给飞行器网络。

[0707] E9. 权利要求E1的计算机实现的方法进一步包括:将该组上行链路命令存储在便携式数据处理系统中。

[0708] E10. 权利要求E1的计算机实现的方法,其中通过已建立的连接从源检索与上行链路命令相对应的一组软件飞行器部件以形成一组检索到的软件飞行器部件包括:通过已建立的连接从源检索一组包中与上行链路命令相对应的一组软件飞行器部件,以形成该组检索到的软件飞行器部件。

[0709] E11. 权利要求E1的计算机实现的方法,其中所述源选自于库和代理服务器应用程序之一。

[0710] E12. 一种设备,包括:能够存储一组命令的数据库;能够存储一组软件飞行器部件的文件系统;能够与选自于库和代理服务器应用程序之一的源通信的库服务;能够直接与飞行器网络中的飞行器网络直接连接的地面组件;管理器,其中该管理器能够通过库服务从地面网络中的源接收该组命令和该组软件飞行器部件、与源断开连接和通过地面组件发送该组命令和该组软件飞行器部件给飞行器网络中的飞行器数据处理系统;以及能够建立与地面网络和飞行器网络的连接的便携式数据处理系统,其中所述库服务、地面组件、数据库和文件系统都位于便携式数据处理系统上。

[0711] E13. 权利要求E12的设备,其中所述管理器包括多个视图,这些视图能够呈现用于启动将该组软件飞行器部件传输给飞行器数据处理系统的图形用户界面。

[0712] E14. 权利要求E13的设备,其中所述多个视图包括上行链路队列视图、上行链路本地清单视图、下传文件视图、事件控制视图和从库检索视图中的至少一个。

[0713] E15. 权利要求E12的设备进一步包括:存储在数据库中的该组命令;以及存储在文件系统上的该组软件飞行器部件。

[0714] E16. 计算机程序产品,包括:计算机可记录存储介质;存储在计算机可记录介质上用于通过地面网络建立便携式数据处理系统与源之间的连接以形成已建立的连接的程序代码;存储在计算机可记录介质上通过已建立的连接从源检索一组上行链路命令的程序代码;存储在计算机可记录介质上通过已建立的连接从源检索与该组上行链路命令相对应的一组软件飞行器部件以形成一组检索到的软件飞行器部件的程序代码;存储在计算机可记录介质上在便携式数据处理系统中存储该组检索到的软件飞行器部件以形成一组存储的软件飞行器部件的程序代码;存储在计算机可记录介质上的断开具有该组存储的软件飞行器部件的便携式数据处理系统与地面网络的连接的程序代码;存储在计算机可记录介质上的将具有该组存储的软件飞行器部件的便携式数据处理系统连接到飞行器中的飞行器网络的程序代码;存储在计算机可记录介质上的通过便携式数据处理系统中的地面组件从该组上行链路命令发出一上行链路命令给飞行器网络的程序代码;以及存储在计算机可记录

介质上的通过地面组件发送与该上行链路命令对应的存储的软件飞行器部件给飞行器网络的程序代码。

[0715] E17. 权利要求E16的计算机程序产品进一步包括: 存储在计算机可记录介质上用于从飞行器网络接收一组下行链路文件以形成一组接收的下行链路文件的程序代码,。

[0716] E18. 权利要求E16的计算机程序产品进一步包括: 存储在计算机可记录介质上用于识别多个软件飞行器部件中的每个软件飞行器部件是否成功地发送给飞行器网络以形成状态的程序代码; 以及存储在计算机可记录介质上用于响应与地面网络建立的后续连接发送状态给源的程序代码。

[0717] E19. 权利要求E16的计算机程序产品, 其中程序代码存储在数据处理系统中的计算机可记录存储介质并且其中程序代码通过网络从远程数据处理系统下载。

[0718] E20. 权利要求E16的计算机程序产品, 其中程序代码存储在服务器数据处理系统中的计算机可记录存储介质的程序代码并且其中程序代码通过网络被下载到远程数据处理系统以通过远程数据处理系统在另一个计算机可记录介质中使用。

[0719] 用于管理软件飞行器部件的计算机实现的方法, 设备和计算机程序产品。通过地面网络在便携式数据处理系统和源之间建立连接。通过已建立的连接从源检索上行链路命令; 通过已建立的连接从源检索与这些上行链路命令相对应的软件飞行器部件。在便携式数据处理系统中存储检索到的软件飞行器部件。断开便携式数据处理系统与地面网络的连接。将便携式数据处理系统连接到飞行器中的飞行器网络。通过便携式数据处理系统中的地面组件从这些上行链路命令发出一上行链路命令给飞行器网络。通过地面组件发送与该上行链路命令对应的存储软件飞行器部件给飞行器网络。

[0720] 机载电子分配系统

[0721] 实施例可以要求权利如下:

[0722] F1. 用于与飞行器传输信息的计算机实现的方法, 该计算机实现的方法包括: 建立在飞行器中的飞行器数据处理系统中执行的机载电子分配系统与地面组件之间的连接; 响应于通过连接做出的来自地面组件的对命令的请求, 识别由机载电子分配系统执行的命令, 以形成已识别的命令; 从地面组件发送已识别的命令给机载电子分配系统; 分派事务处理标识符给该命令; 使用事务处理标识符维持事务处理的状态, 该事务处理与机载电子分配系统和地面组件上的该命令关联; 由地面组件启动上行链路; 从地面组件发送软件飞行器部件给机载电子分配系统以进行上传; 以及在地面组件上保存软件飞行器部件的传输状态。

[0723] F2. 权利要求F1的计算机实现的方法进一步包括: 响应将从机载电子分配系统发送给地面组件的下行链路文件发送给地面组件的另一个请求, 确定是否发送下行链路文件; 响应发送下行链路文件给地面组件的决定而发送下行链路文件给地面组件。

[0724] F3. 权利要求F1的计算机实现的方法, 其中发送软件飞行器部件的步骤包括: 如果状态指出存在软件飞行器部件的未发送部分, 则发送软件飞行器部件的未发送部分给机载电子分配系统。

[0725] F4. 权利要求F2的计算机实现的方法, 其中发送下行链路文件给地面组件的步骤包括: 发送下行链路文件的未发送部分给地面组件; 将由地面组件接收的下行链路文件的未发送部分附加到对应下行链路文件的部分下行链路文件。

[0726] F5. 权利要求F4的计算机实现的方法进一步包括:使用下行链路文件的相应传输状态识别下行链路文件的未发送部分。

[0727] F6. 权利要求F1的计算机实现的方法进一步包括:将软件飞行器部件存储在飞行器数据处理系统中的存储装置中,机载电子分配系统位于该飞行器数据处理系统上。

[0728] F7. 权利要求F2的计算机实现的方法,其中下行链路文件包括飞行器中的外场可更换单元生成的数据。

[0729] F8. 权利要求F7的计算机实现的方法,其中所述外场可更换单元选自于飞行管理系统、自动驾驶仪、飞行娱乐系统、通信系统、导航系统、飞行控制器、飞行记录器和碰撞避免系统之一。

[0730] F9. 权利要求F1的计算机实现的方法,其中所述软件飞行器部件被用于飞行器中的外场可更换单元。

[0731] F10. 计算机程序产品,包括:计算机可读介质;存储在计算机可读介质上用于建立在飞行器中的飞行器数据处理系统中执行的机载电子分配系统和地面组件之间的连接的程序代码;存储在计算机可读介质上响应于来自机载电子分配系统通过连接做出的对命令的请求,识别由机载电子分配系统执行的命令以形成已识别的命令的程序代码;存储在计算机可读介质上从地面组件发送已识别的命令给机载电子分配系统的程序代码;存储在计算机可读介质上分派事务处理标识符给该命令的程序代码;存储在计算机可读介质上使用事务处理标识符维持事务处理的状态的程序代码,该事务处理与机载电子分配系统和地面组件上的该命令关联;存储在计算机可读介质上用于由机载电子分配系统启动上行链路的程序代码;存储在计算机可读介质上从地面组件发送软件飞行器部件给机载电子分配系统以进行上传的程序代码;以及存储在计算机可读介质上用于保存软件飞行器部件的传输状态的程序代码。

[0732] F11. 权利要求F10的计算机程序产品进一步包括:存储在计算机可读介质上的程序代码,用于响应将从机载电子分配系统发送给地面组件的下行链路文件发送给地面组件的另一个请求,判断是否发送下行链路文件;存储在计算机可读介质上的程序代码,用于响应判断为发送下行链路文件给地面组件而发送下行链路文件给地面组件。

[0733] F12. 权利要求F10的计算机程序产品,其中存储在计算机可读介质上用于从地面组件发送软件飞行器部件给机载电子分配系统以进行上传的程序代码包括:存储在计算机可读介质上的程序代码,如果状态指出存在软件飞行器部件的未发送部分,则该程序代码用于发送软件飞行器部件的未发送部分给机载电子分配系统。

[0734] F13. 权利要求F11的计算机程序产品,其中存储在计算机可读介质上用于发送下行链路文件给地面组件的程序代码包括:存储在计算机可读介质上用于发送下行链路文件的未发送部分给地面组件的程序代码;存储在计算机可读介质上用于将由地面组件接收的下行链路文件的未发送部分附加到对应于该下行链路文件的部分下行链路文件的程序代码。

[0735] F14. 权利要求F13的计算机程序产品进一步包括:存储在计算机可读介质上用于使用下行链路文件的相应的传输状态识别下行链路文件的未发送部分的程序代码。

[0736] F15. 权利要求F10的计算机程序产品进一步包括:存储在计算机可读介质上用于在其上置有机载电子分配系统的飞行器数据处理系统中的存储装置中存储软件飞行器部

件的程序代码。

[0737] F16. 权利要求F11的计算机程序产品,其中下行链路文件包括有飞行器中的外场可更换单元生成的数据。

[0738] F17. 一种设备,包括:地面组件;机载电子分配系统,其能够从地面组件接收命令,响应从地面组件接收上行链路命令,请求对应于上行链路命令的软件飞行器部件,响应对该软件飞行器部件的请求从地面组件接收软件飞行器部件以形成接收的软件飞行器部件;以及存储软件飞行器部件;数据处理系统,其中地面组件在数据处理系统上执行;以及飞行器数据处理系统,其中机载电子分配系统在飞行器数据处理系统上执行。

[0739] F18. 权利要求F17的设备,其中机载电子分配系统能够将发送下行链路文件给地面组件的请求发送给地面组件并且响应该地面组件能够接收下行链路文件的指示,发送下行链路文件给地面组件。

[0740] F19. 权利要求F17的设备进一步包括:飞行器中的飞行器网络,其中飞行器数据处理系统位于飞行器网络中。

[0741] F20. 权利要求F17的设备进一步包括:飞行器中的一组外场可更换单元,其中软件飞行器部件用于飞行器中的该组外场可更换单元中的外场可更换单元之一。

[0742] 计算机实现的方法、设备和计算机程序产品,用于与飞行器传输信息。建立在飞行器中的飞行器数据处理系统中执行的机载电子分配系统与地面组件之间的连接。响应来自地面组件的对命令的请求,识别用于执行的命令。已识别的命令从地面组件发送给机载电子分配系统。分配事务处理标识符给命令。使用事务处理标识符将与命令相关联的事务处理保持在机载电子分配系统和地面组件上。由地面组件启动上行链路。从地面组件发送软件飞行器部件给机载电子分配系统以进行上传。保存在地面组件上的软件飞行器部件的传输状态。

[0743] 软件飞行器部件库

[0744] 实施例可要求权利如下:

[0745] G1. 用于管理软件飞行器部件的计算机实现的方法,该计算机实现的方法包括:在库接收软件飞行器部件以形成接收的软件飞行器部件;将接收的软件飞行器部件存储在具有多个软件飞行器部件的库中的存储器中;提供用户界面系统以操控库中的多个软件飞行器部件;响应从用户界面系统接收用户输入以从库中的多个软件飞行器部件上传所选的软件飞行器部件给目标飞行器数据处理系统,创建命令以发送所选的软件飞行器部件给在其中置有目标飞行器数据处理系统的飞行器以形成上行链路命令;将上行链路命令存储到库中的存储器中的队列中以形成存储的上行链路命令;响应来自代理服务器应用程序的对命令的请求,分配存储的上行链路命令给代理服务器应用程序;以及响应接收来自处理所存储的上行链路命令的代理服务器应用程序的对所选软件飞行器部件的请求,发送所选的软件飞行器部件给代理服务器应用程序。

[0746] G2. 权利要求G1的计算机实现的方法,其中分配步骤包括:响应于存在与代理服务器应用程序的连接,为代理服务器应用程序识别上行链路命令;创建上行链路命令的命令数据结构;发送命令数据结构给代理服务器应用程序,其中代理服务器应用程序使用该命令数据结构开始处理命令;以及响应在发送命令数据结构之后接收的对上行链路命令的另一个请求,发送上行链路命令给代理服务器应用程序。

[0747] G3. 权利要求G1的计算机实现的方法进一步包括:响应来自软件管理工具的对命令的请求,分配存储的上行链路命令给软件管理工具。

[0748] G4. 权利要求G1的计算机实现的方法进一步包括:将软件飞行器部件存储在存储器装置中的包中。

[0749] G5. 权利要求G4的计算机实现的方法,其中软件飞行器部件被数字签名并且包被数字签名。

[0750] G6. 权利要求G1的计算机实现的方法,其中软件飞行器部件被存储在存储器中的文件系统中并且其中队列位于存储器中的数据库中。

[0751] G7. 权利要求G1的计算机实现的方法进一步包括:响应另一个用户输入,从多个软件飞行器部件删除一组飞行器部件。

[0752] G8. 权利要求G1的计算机实现的方法进一步包括:接收关于软件飞行器部件的上行链路的状态信息以形成接收的状态信息;并且存储接收的状态信息。

[0753] G9. 一种设备,包括:位于存储器装置上的文件系统,其中该文件系统存储多个软件飞行器部件;位于该存储器装置上的数据库,其中数据库存储被用于管理多个软件飞行器部件的多个命令;能够生成多个命令的管理组件;以及能够接收用户输入以操作管理组件的用户界面系统。

[0754] G10. 权利要求G9的设备,其中所述管理组件包括部件室和命令调度器。

[0755] G11. 权利要求G9的设备,其中所述存储器装置是一组硬盘驱动器。

[0756] G12. 一种计算机,包括:总线;连接到总线的通信适配器;连接到总线的存储器,其中计算机可用的程序代码被存储在总线上;以及连接到总线的处理器单元,其中处理器单元执行计算机可用的程序代码以在库接收软件飞行器部件,从而形成接收的软件飞行器部件;将接收的软件飞行器部件存储在具有多个软件飞行器部件的库中的存储器中;提供用户界面系统以操控库中的多个软件飞行器部件;响应从用户界面系统接收上传从库中的多个软件飞行器部件中选出的软件飞行器部件给目标飞行器数据处理系统的用户输入,创建命令以发送所选的软件飞行器部件给在其中置有目标飞行器数据处理系统的飞行器,从而形成上行链路命令;将上行链路命令存储到库中的存储器中的队列中以形成存储的上行链路命令;响应来自代理服务器应用程序的对命令的请求,分配存储的上行链路命令给代理服务器应用程序;以及响应从处理所存储的上行链路命令的代理服务器应用程序接收的对所选软件飞行器部件的请求,发送所选的软件飞行器部件给代理服务器应用程序。

[0757] G13. 权利要求G12的计算机,其中在执行计算机可用程序代码以响应来自代理服务器应用程序对命令的请求,分配存储的上行链路命令给代理服务器应用程序时,处理器单元执行计算机可用程序代码以响应存在与的代理服务器应用程序的连接,为代理服务器应用程序识别上行链路命令;创建上行链路命令的命令数据结构;发送命令数据结构给代理服务器应用程序,其中代理服务器应用程序使用该命令数据结构开始处理命令;以及响应对在发送命令数据结构之后接收的上行链路命令的另一个请求发送上行链路命令给代理服务器应用程序。

[0758] G14. 权利要求G12的计算机,其中,处理器单元进一步执行计算机可用程序代码以响应来自软件管理工具的对命令的请求,分配存储的上行链路命令给软件维护工具。

[0759] G15. 计算机程序产品包括:计算机可记录存储介质;存储在计算机可记录存储介

质上用于在库接收软件飞行器部件以形成接收的软件飞行器部件的程序代码;存储在计算机可记录存储介质上用于将接收的软件飞行器部件存储在具有多个软件飞行器部件的库中的存储器中的程序代码;存储在计算机可记录存储介质上用于提供用户界面系统以操控库中的多个软件飞行器部件的程序代码;存储在计算机可记录存储介质上用于响应从用户界面系统接收用户输入以从库中的多个软件飞行器部件上传所选的软件飞行器部件给目标飞行器数据处理系统,创建命令以发送所选的软件飞行器部件给在其中置有目标飞行器数据处理系统的飞行器以形成上行链路命令的程序代码;存储在计算机可记录存储介质上用于将上行链路命令存储到库中的存储器中的队列中以形成存储的上行链路命令的程序代码;存储在计算机可记录存储介质上用于响应来自代理服务器应用程序的对命令的请求,分配存储的上行链路命令给代理服务器应用程序的程序代码;以及存储在计算机可记录存储介质上用于响应接收来自处理所存储的上行链路命令的代理服务器应用程序的对所选软件飞行器部件的请求,发送所选的软件飞行器部件给代理服务器应用程序的程序代码。

[0760] G16. 权利要求15的计算机程序产品,其中,存储在计算机可记录存储介质上用于响应来自代理服务器应用程序的对命令的请求,分配存储的上行链路命令给代理服务器应用程序的程序代码包括:存储在计算机可记录存储介质上用于响应存在和代理服务器应用程序的连接,为代理服务器应用程序识别上行链路命令的程序代码;存储在计算机可记录存储介质上用于创建上行链路命令的命令数据结构的程序代码;存储在计算机可记录存储介质上用于发送命令数据结构给代理服务器应用程序的程序代码,其中代理服务器应用程序使用该命令数据结构开始处理命令;以及存储在计算机可记录存储介质上用于响应在发送命令数据结构之后接收的对上行链路命令的另一个请求,发送上行链路命令给代理服务器应用程序的程序代码。

[0761] G17. 权利要求15的计算机程序产品进一步包括:存储在计算机可记录存储介质上用于响应来自软件管理工具对命令的请求分配存储的上行链路命令给软件管理工具的程序代码。

[0762] G18. 权利要求15的计算机程序产品进一步包括:存储在计算机可记录存储介质上用于将软件飞行器部件存储在存储装置中的包中的程序代码。

[0763] G19. 权利要求18的计算机程序产品,其中软件飞行器部件被数字签名并且包被数字签名。

[0764] G20. 权利要求15的计算机程序产品,其中软件飞行器部件被存储在存储器中的文件系统中并且其中队列位于存储器中的数据库中。

[0765] G21. 权利要求15的计算机程序产品进一步包括:存储在计算机可记录存储介质上用于响应其他用户输入从多个软件飞行器部件中删除一组软件飞行器部件的程序代码。

[0766] G22. 权利要求15的计算机程序产品进一步包括:存储在计算机可记录存储介质上用于接收关于软件飞行器部件的上行链路的状态信息以形成接收的状态信息的程序代码;以及存储在计算机可记录存储介质上用于存储接收的状态信息的程序代码。

[0767] 计算机实现的方法、设备和计算机程序产品,用于管理软件飞行器部件。在库接收软件飞行器部件。将接收的软件飞行器部件存储在具有多个软件飞行器部件的所述库中。提供用户界面系统以操控软件飞行器部件。响应从用户界面接收上传所选的软件飞行器部

件给目标飞行器数据处理系统的用户输入,创建命令以发送所选的软件飞行器部件给在其中置有目标飞行器数据处理系统的飞行器。将上行链路命令存储到库中的存储器中的队列中。响应对命令的请求,分配上行链路命令给代理服务器应用程序。响应从处理存储的上行链路命令的代理服务器应用程序接收的请求,发送所选部件给代理服务器应用程序。

[0768] 给出所描述的不同有利实施例用于说明和描述的目的,并且并不试图穷举或限制实施例为公开的形式。许多修改和变化对于本领域普通技术人员来说是显然的。另外,不同的有利实施例可以提供与其他有利实施例的比较得出的不同优势。选择和描述所选的一个或多个实施例,其目的是最佳地解释实施例的原理、实际应用,并且使得本领域一般技术人员能够理解针对不同实施例的公开,这些实施例具有不同的修改从而适合特定的使用的目的。

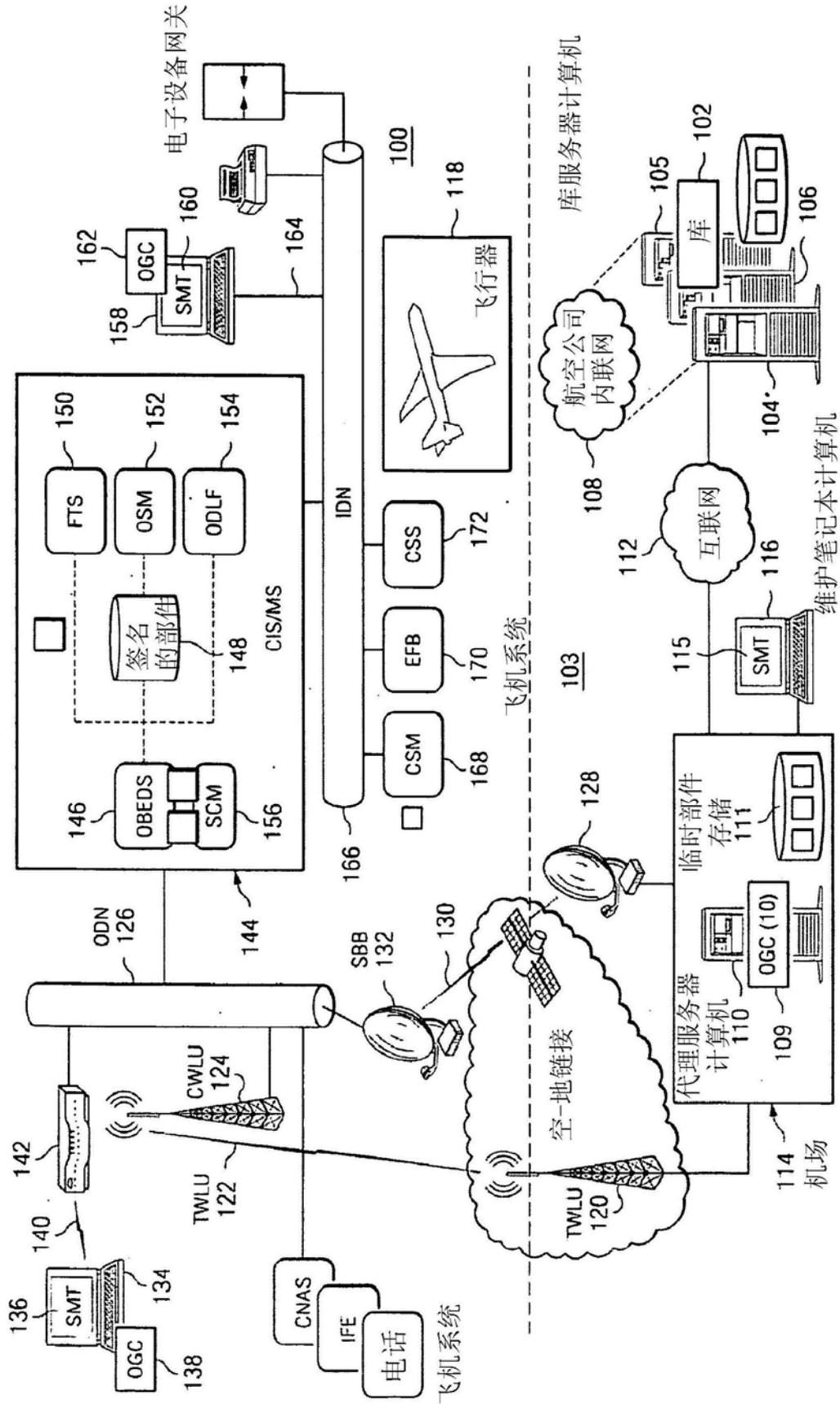


图1

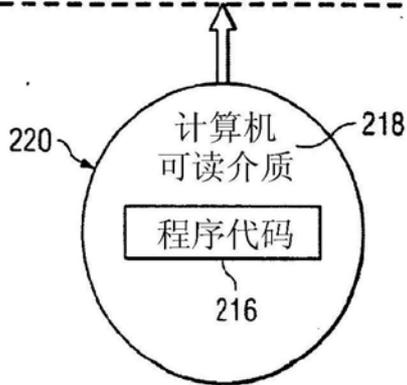
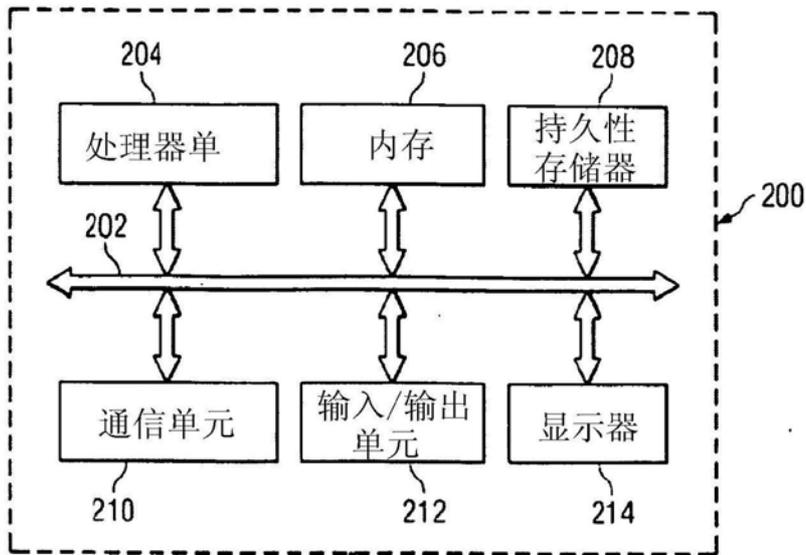


图2

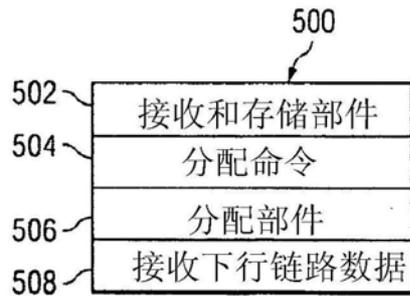


图5

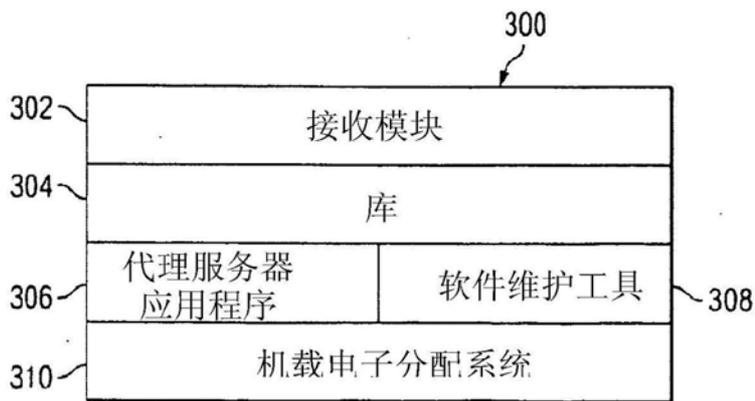


图3

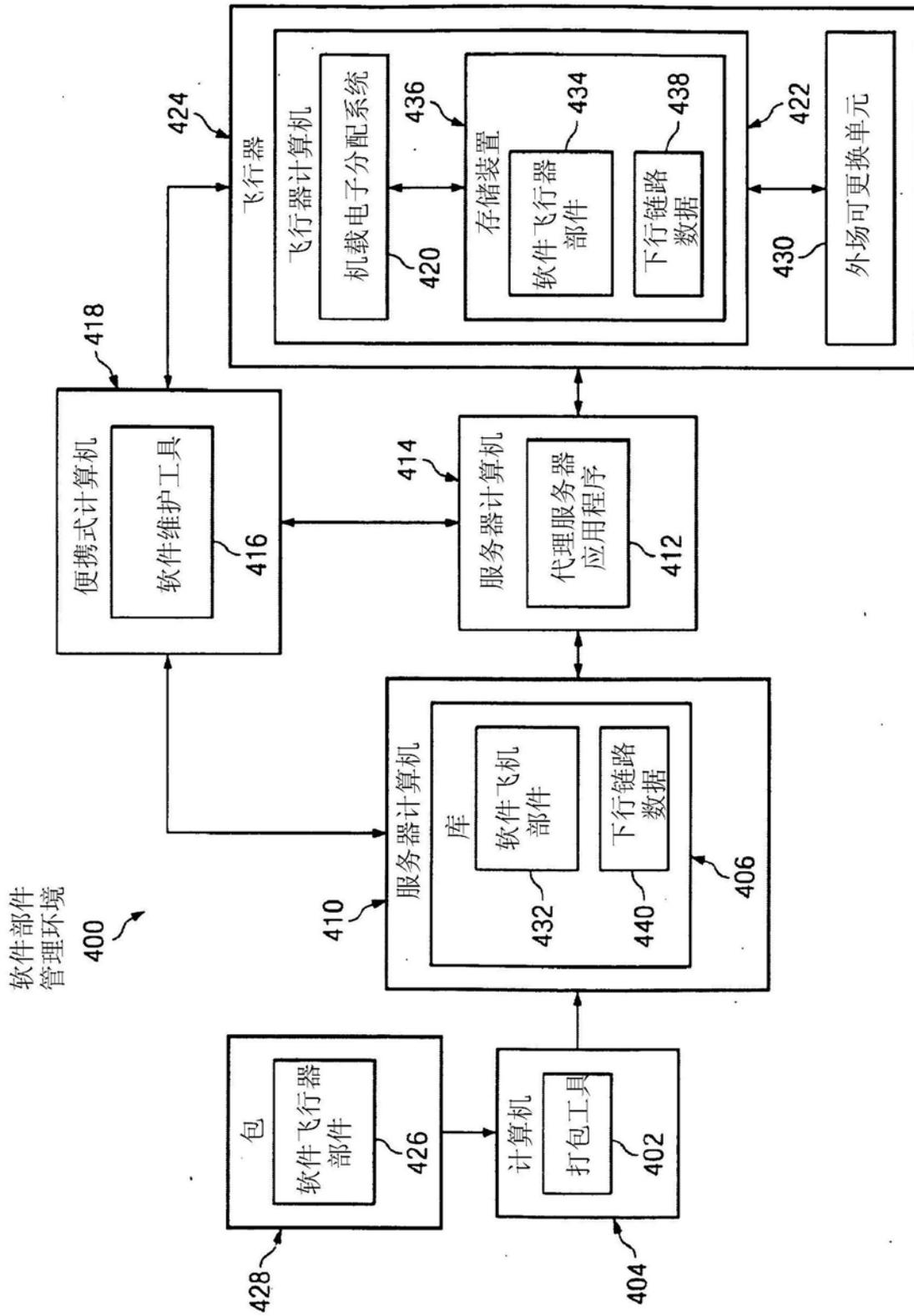


图4

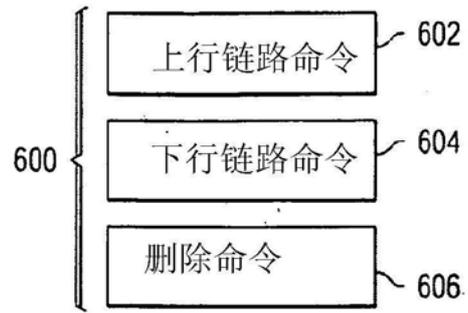


图6

700
↓

```
<Cmd xmlns="http://www.boeing.com/787/obeds">  
  <MsgId>kmad09ce87i6ef3c</MsgId> ~ 702  
  <Type>uplink</Type> ~ 704  
  <System>CIS-MS</System> ~ 706  
  <Appld>OSM</Appld> ~ 708  
  <LinkLabel>SMT</LinkLabel> ~ 710  
  <ServerAddress>172.24.10.50</ServerAddress> ~ 712  
  <DataType>LSAP</DataType> ~ 714  
  <Resource type="LSAP">PNU 120 OJSDO</Resource> ~ 716  
</Cmd>
```

图7

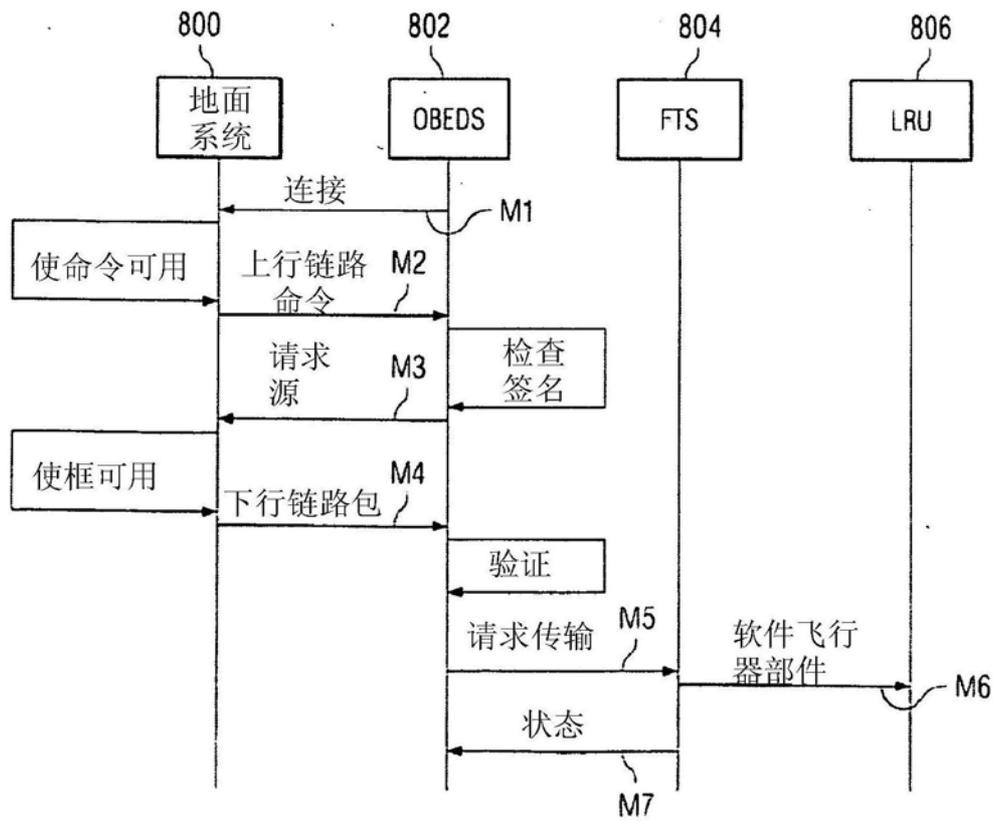


图8

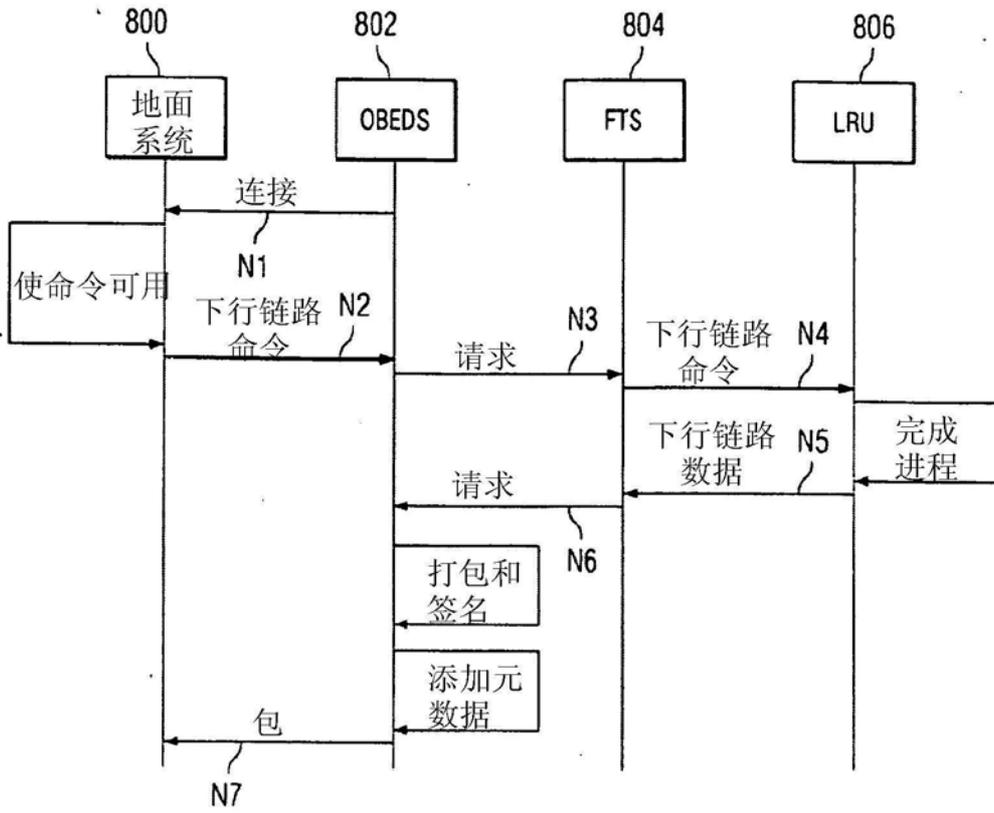


图9

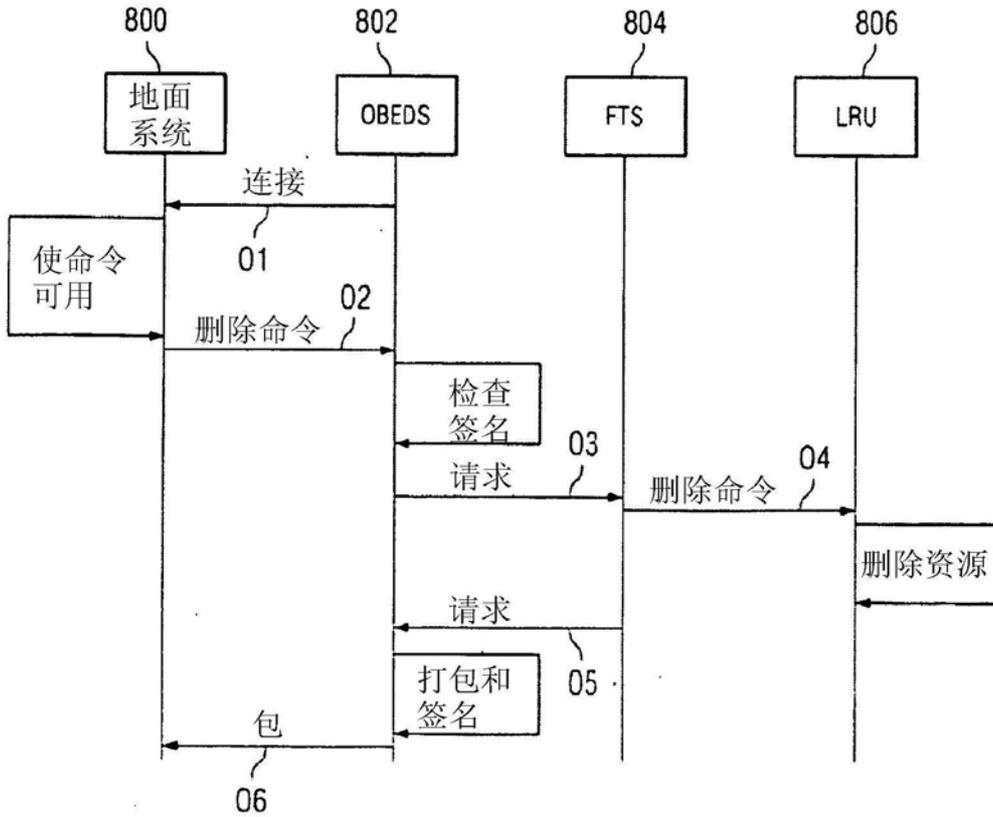


图10

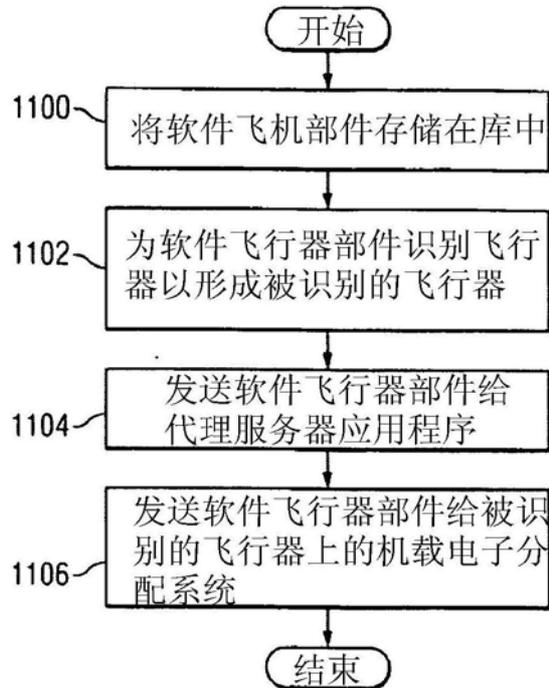


图11

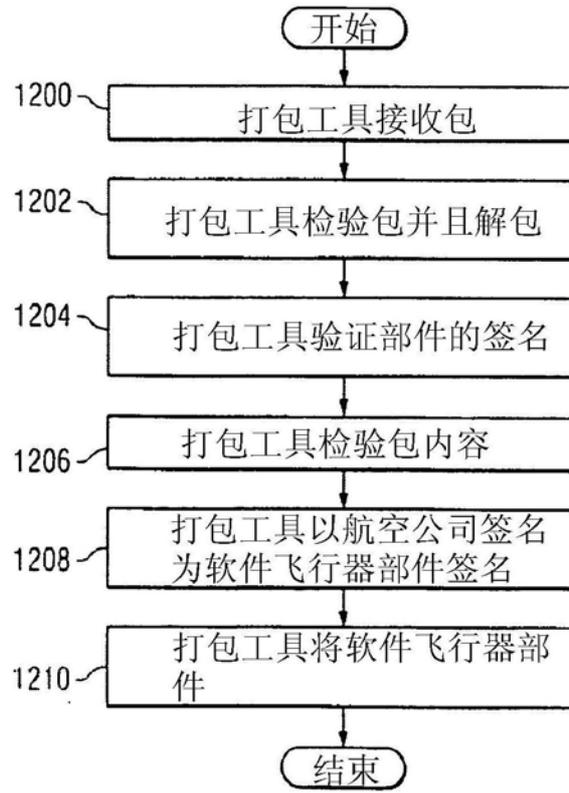


图12

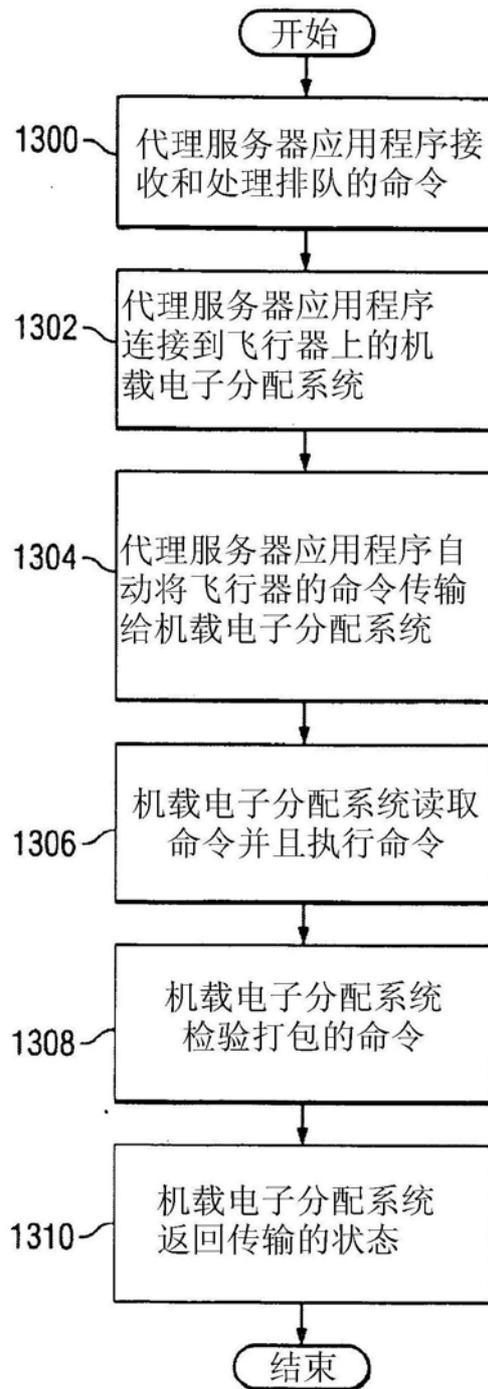


图13

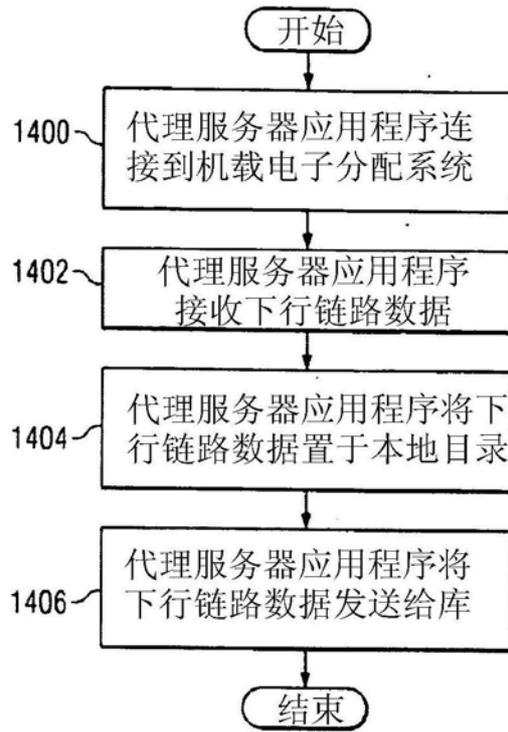


图14

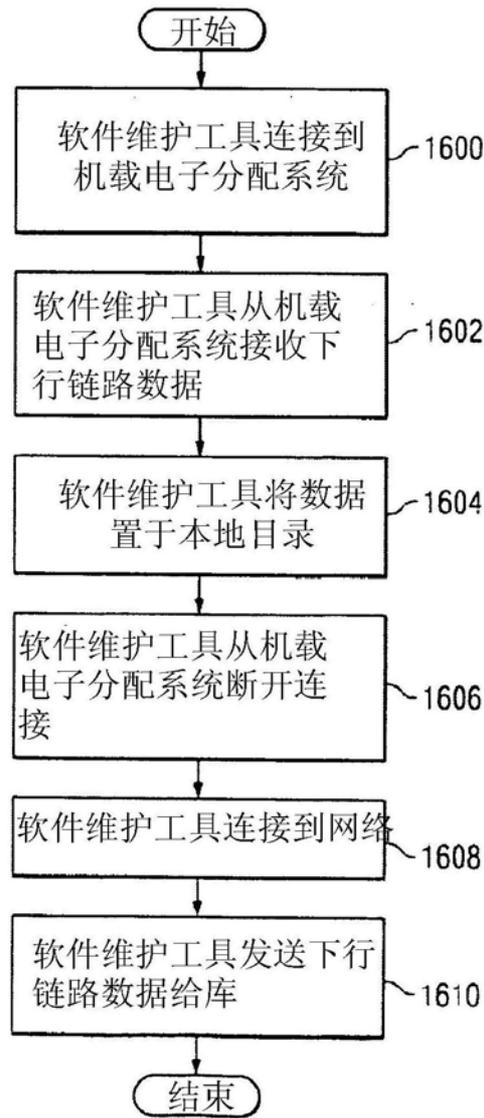


图16

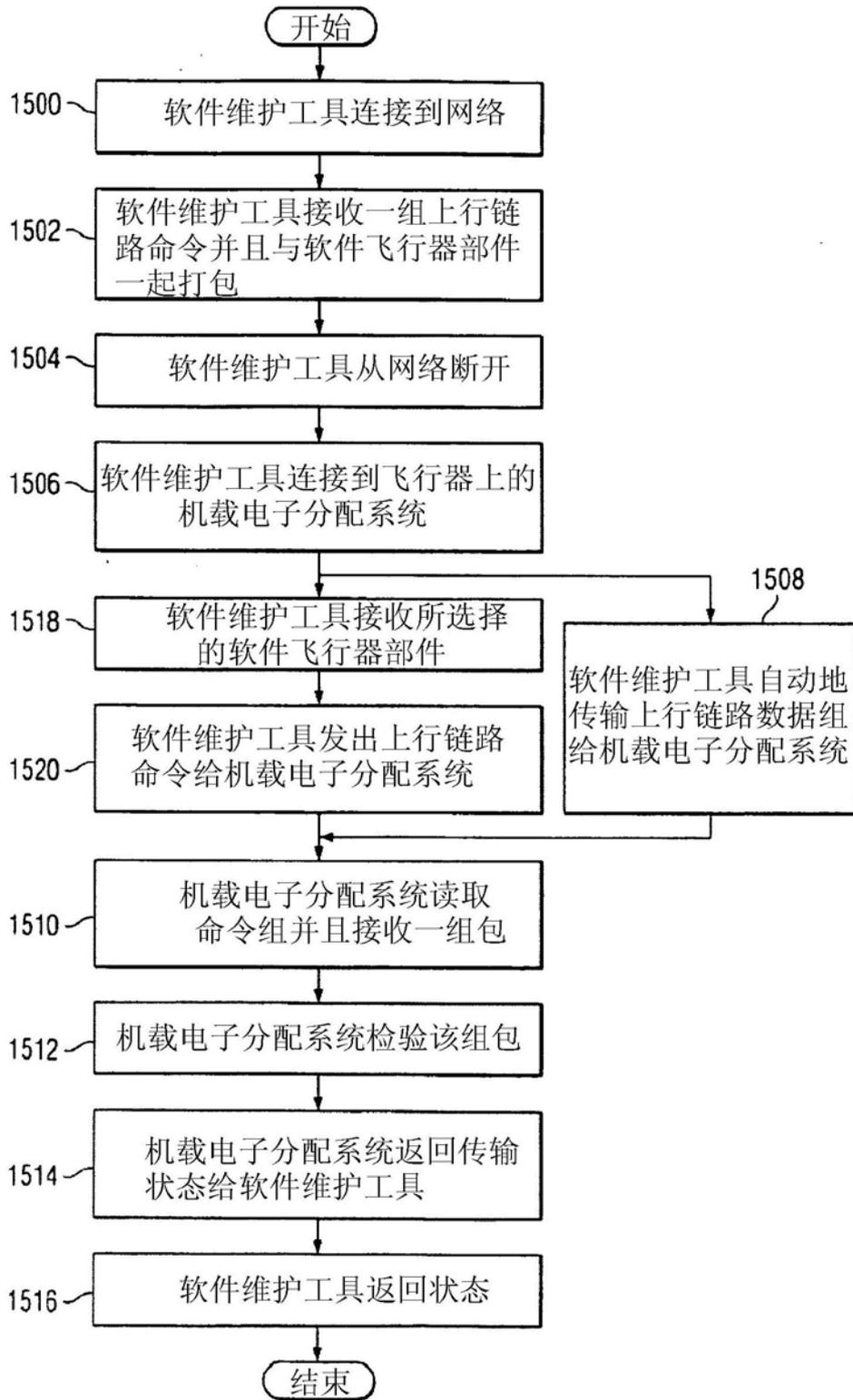


图15

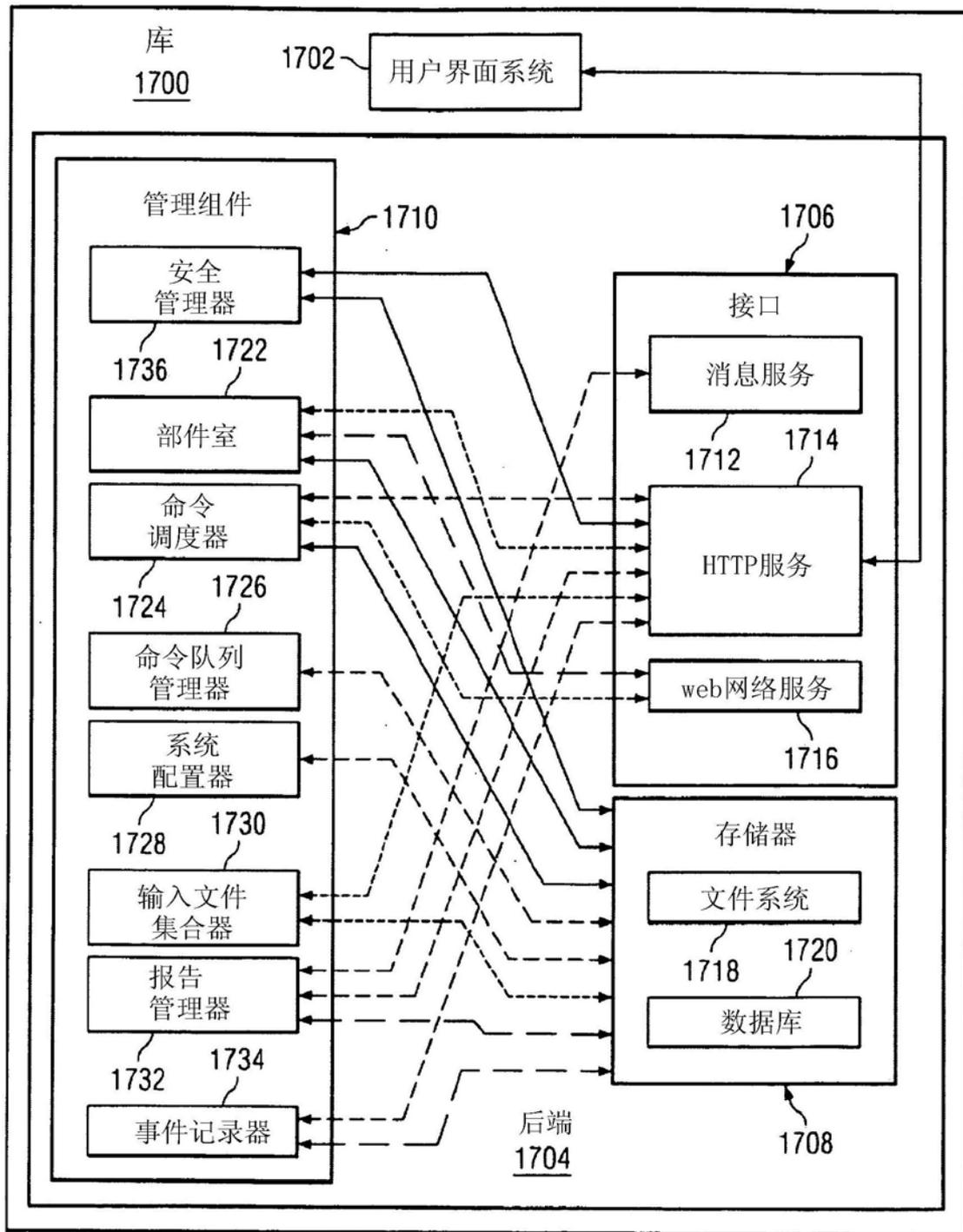


图17

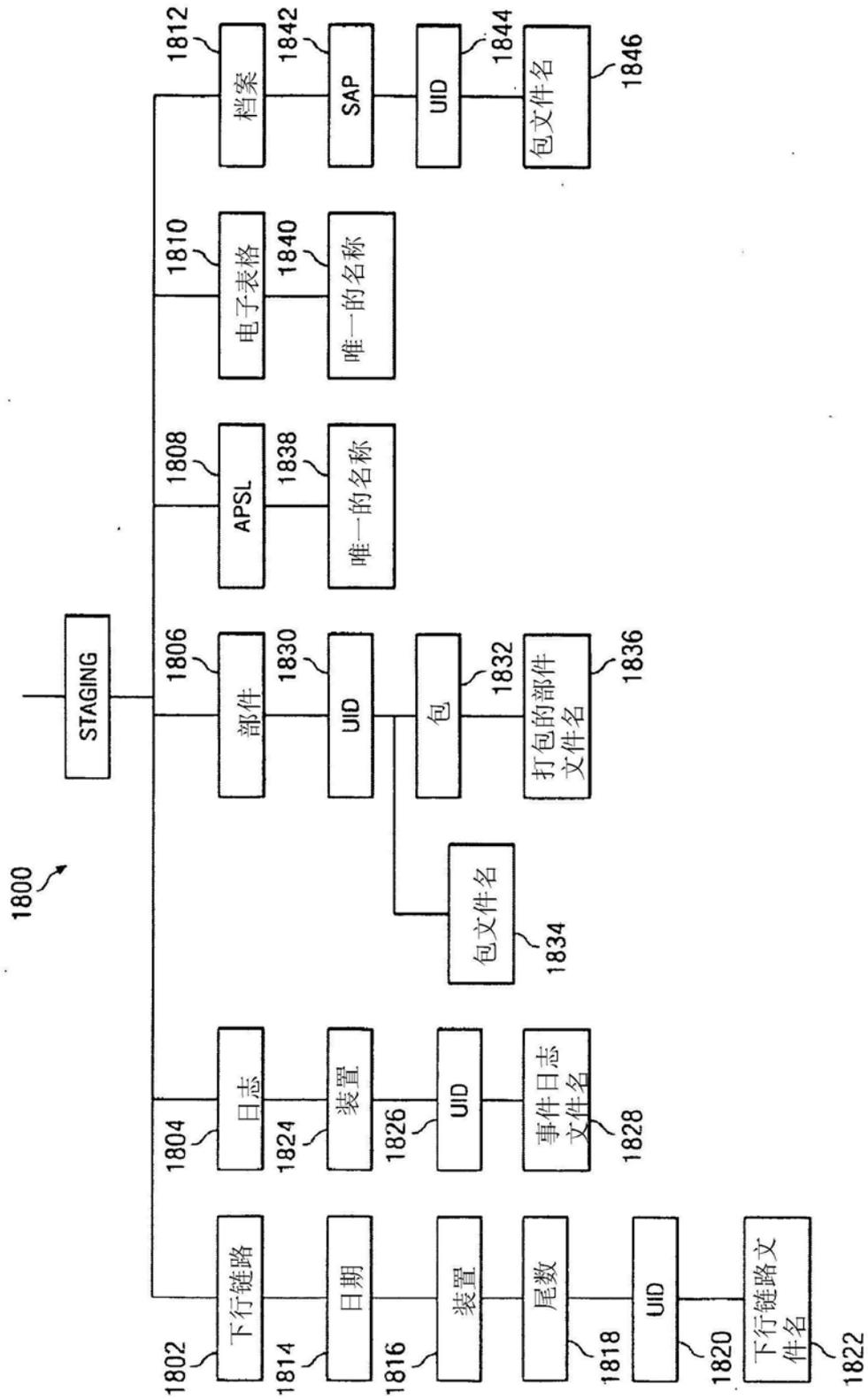


图18

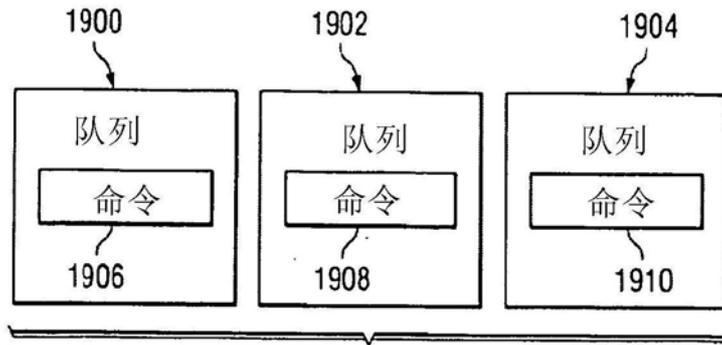


图19

图19

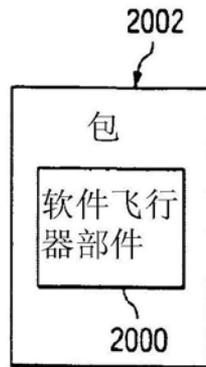


图20

```

Delete Command:
//A list of file name to be deleted
String[] fileNames; ~ 2102

//Or a list of part ID to be deleted
long [] partIds; ~ 2104

//A list of airplane ID receiving command
long [] airplaneIds; ~ 2106

//A list of device ID to serve the command
long [] deviceIds; ~ 2108

//the destination system to receive the command
long sysAppId; ~ 2110

//device ID that request the command.
String user; ~ 2112

```

图21

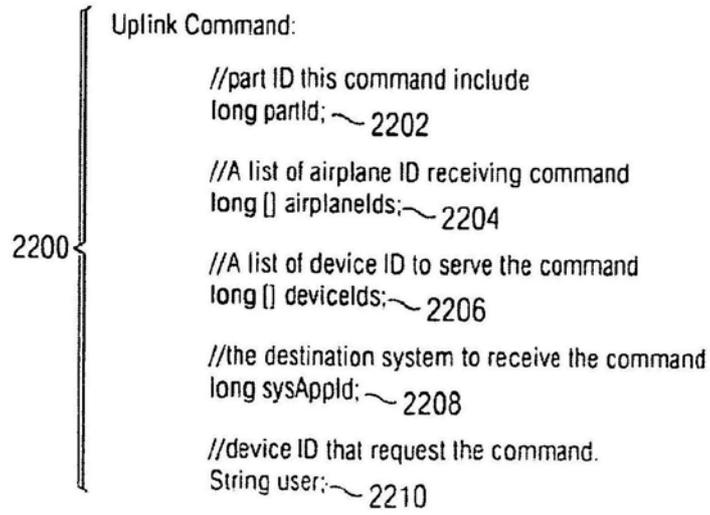


图22

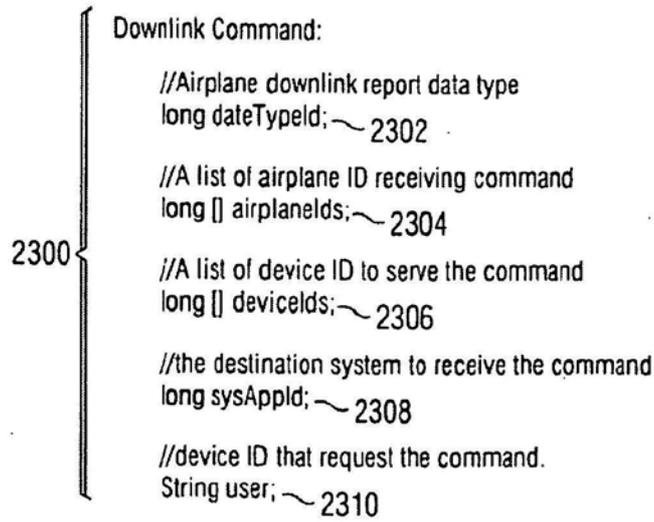


图23

2400

Command Queues

Command Dispatcher

上行链路
下行链路 2402
删除 2404
2406

Uplink Commands ? 2408 [Reset] [Refresh] [Dispatch]

Airplane Tail #	Proxy Server(s)	SMT(s)
<input type="checkbox"/> All	<input type="checkbox"/> All <input type="checkbox"/> None	<input type="checkbox"/> All <input type="checkbox"/> None
B007	DEVPSTACK	LSAPL SMT
B123	PARIS	MINNEAPOLIS
B3422	PS_A3841707	SMT 123
Boeing123	PS_DEFAULT	SMT Seattle
Boeing295	PS_Device1	SMT super
Boeing456 2410	PS_MIKE 2412	SMT1 2414
Boeing581	PS_PETER2	SMT2
Boeing789	PS_SEATTLE	SMTSeattle
Boeing928	SEATTLE 2416	SMT_Device1
J1234	testDevice1	SMT_Device2

2418 Destination System Airplane Condition Monitoring Function (ACM)

Part Item XXX-Z-A000-0100 >> 2426 2428

IP Owner XXX82 2420 Product Status BLUE_LABEL

Applicable STD. 140972 Cert. Expiration Date ARINC665-2

2422

2424 XXX82-A000-0100

Parts Vault

Maintenance

System Services

Reports

图24

Command Queues

未决的命令
已执行的命令
出队列的命令

2506 2504 2502

记录命令
2508

PENDING COMMANDS

ORDER*	TAIL#*	CMD TYPE*	GROUP ID*	CMD ID*	CREATE TIME*	DE
1437	83422	UPUNK	936_B3422	9cd91d8cccb082e15b51ce819b0ee9	Oct 02, 2007 09:38:04	SMT2
1436	8123	UPLINK	936_B123	33e89976dc866d1d410cc52c59bd65b	Oct 02, 2007 09:38:04	SMT2
1434	83422	UPLINK	935_B3422	8cb4ef1433556e0e713072e45b03bb98	Oct 02, 2007 09:37:35	SMT2
1433	8123	UPLINK	935_B123	861d661e817393e12d894e9603c0c24f	Oct 02, 2007 09:37:35	SMT2
1431	83422	UPLINK	934_B3422	7311d414394db0615ab31e1742dc4d0	Oct 02, 2007 09:37:20	SMT2
1430	8123	UPLINK	934_B123	83cd0601cc06892e9b86559cc8de59ad	Oct 02, 2007 09:37:20	SMT2
1428	YAL9876	DOWNLINK	933_YAL9876	36c9696a21443d243443ed53d841eea	Sep 25, 2007 17:55:40	PS_Device
1427	YAL3579	DOWNLINK	933_YAL3579	e1701e735713d10b360b1978ef28221	Sep 25, 2007 17:55:40	PS_Device
1426	YAL2468	DOWNLINK	933_YAL2468	fc9761e99e8a37e3920e25443377388	Sep 25, 2007 17:55:40	PS_Device
1425	YAL1234	DOWNLINK	933_YAL1234	7e9089c5295701898d1d4f447839d66	Sep 25, 2007 17:55:40	PS_Device
1424	YAL9876	DELETE	932_YAL9876	119500f55ed4be117ce618507b146cdf	Sep 25, 2007 17:54:34	PS_Device
1423	YAL3579	DELETE	932_YAL3579	ba461d6622f14b7946c101cblde60b1	Sep 25, 2007 17:54:34	PS_Device
1422	YAL2468	DELETE	932_YAL2468	e456213752ba7deed4393865171c15ba	Sep 25, 2007 17:54:34	PS_Device
1421	YAL1234	DELETE	932_YAL1234	b60ef1e3d1710070ab01c941c4bb3de5	Sep 25, 2007 17:54:34	PS_Device
1420	YAL9876	DELETE	931_YAL9876	a70143eeed85e874005672428339b536	Sep 25, 2007 17:53:52	PS_Device
1419	YAL3579	DELETE	931_YAL3579	7df860dc00400021f89d5cc0e0ed02b5	Sep 25, 2007 17:53:52	PS_Device
1418	YAL2468	DELETE	931_YAL2468	f55e700b808d2556e1192cc63db6978	Sep 25, 2007 17:53:52	PS_Device
1417	YAL1234	DELETE	931_YAL1234	d6defdc37139115cb241914f33ce8d1	Sep 25, 2007 17:53:52	PS_Device
1412	YAL9876	DOWNLINK	929_YAL9876	b1d6c05a7c6444450716071bbbe3b14e	Sep 25, 2007 17:52:40	PS_Device
1411	YAL3579	DOWNLINK	929_YAL3579	68861103b9c31cc5c86811444c81e7db	Sep 25, 2007 17:52:40	PS_Device

Command Dispatcher

Parts Vault

Maintenance

System Services

Reports

2510

2500

图25

Command Queues		VALID PARTS		2610	
Command Dispatcher		NOMENCLATURE	PRODUCTION	EXPIRATION DATE	APPLICABLE STD
Parts Vault					
有效部件	ABC-Z-DXF-123-4 This is a short desc.	RED_LABEL	Jul 13, 2034 17:38:43	AppSTD X	3613
引入部件	DON-Z-123-456-0 This is a short desc.	RED_LABEL	Jul 13, 2034 17:38:43	AppSTD X	2809
过期的部件	XXX-Z-4000-0100 XYX82-4000-0-100	BLUE_LABEL	Jul 13, 2034 17:38:43	ARINC 665-2	140972
故障部件	GN3-987-65-432 This is a rather short descri...	RED_LABEL	Jul 13, 2034 17:38:43	AppSTD X	3532
2608	ENG48-CD50-5003 ZYCVZCY	BLACK_LABEL	Jul 13, 2034 17:38:43	ARINC 665-2	608448
2606	XXXZA-5638-031H This is a rather short descri...	RED_LABEL	Jan 20, 2020 14:55:16	AppSTD X	7950130
	XXXZA-3715-706E This is a rather short descri...	RED_LABEL	Jan 20, 2020 14:55:16	AppSTD X	18046
	XXXZA-5256-907M This is a rather short descri...	RED_LABEL	Jan 20, 2020 14:55:16	AppSTD X	939443
	XXXZA-9606-909K This is a rather short descri...	RED_LABEL	Jan 20, 2020 14:55:16	AppSTD X	186860
	XXXZA-0690-170K This is a rather short descri...	RED_LABEL	Jan 20, 2020 14:55:16	AppSTD X	547803
	XXXZA-9479-657E This is a rather short descri...	RED_LABEL	Jan 20, 2020 14:55:16	AppSTD X	25727
	XXXZA-6807-071M This is a rather short descri...	RED_LABEL	Jan 20, 2020 14:55:16	AppSTD X	7949896
	ARI-Z665LSP ASDFA	BLUE_LABEL	Jul 13, 2034 17:38:43	ARINC 665-2	4770
	TTTTTTTTTTTT ZCV	BLACK_LABEL	Jul 13, 2034 17:38:43	ARINC 665-2	2967
	BCC22-0000-001U BCC22-0000-001U	BLACK_LABEL	Jul 13, 2034 17:38:43	ARINC 665-2	234504
	XXXZA-7866-203J This is a rather short descri...	RED_LABEL	Jan 20, 2020 14:55:16	AppSTD X	3381318
	HNP5A-AL01-100I This is a rather short descri...	RED_LABEL	Jul 13, 2034 17:38:43	AppSTD X	4770
	ASDFASDF	BLACK_LABEL	Jul 13, 2034 17:38:43	ARINC 665-2	5094
	MHARHISH	BLACK_LABEL	Jun 18, 2015 16:59:59	ARINC 665-2	386533
	HEY-ZEREGUYS111 ASDF	BLUE_LABEL	Jul 13, 2034 17:38:43	ARINC 665-2	4776
Maintenance					
System Services					
Reports					

2600

图26

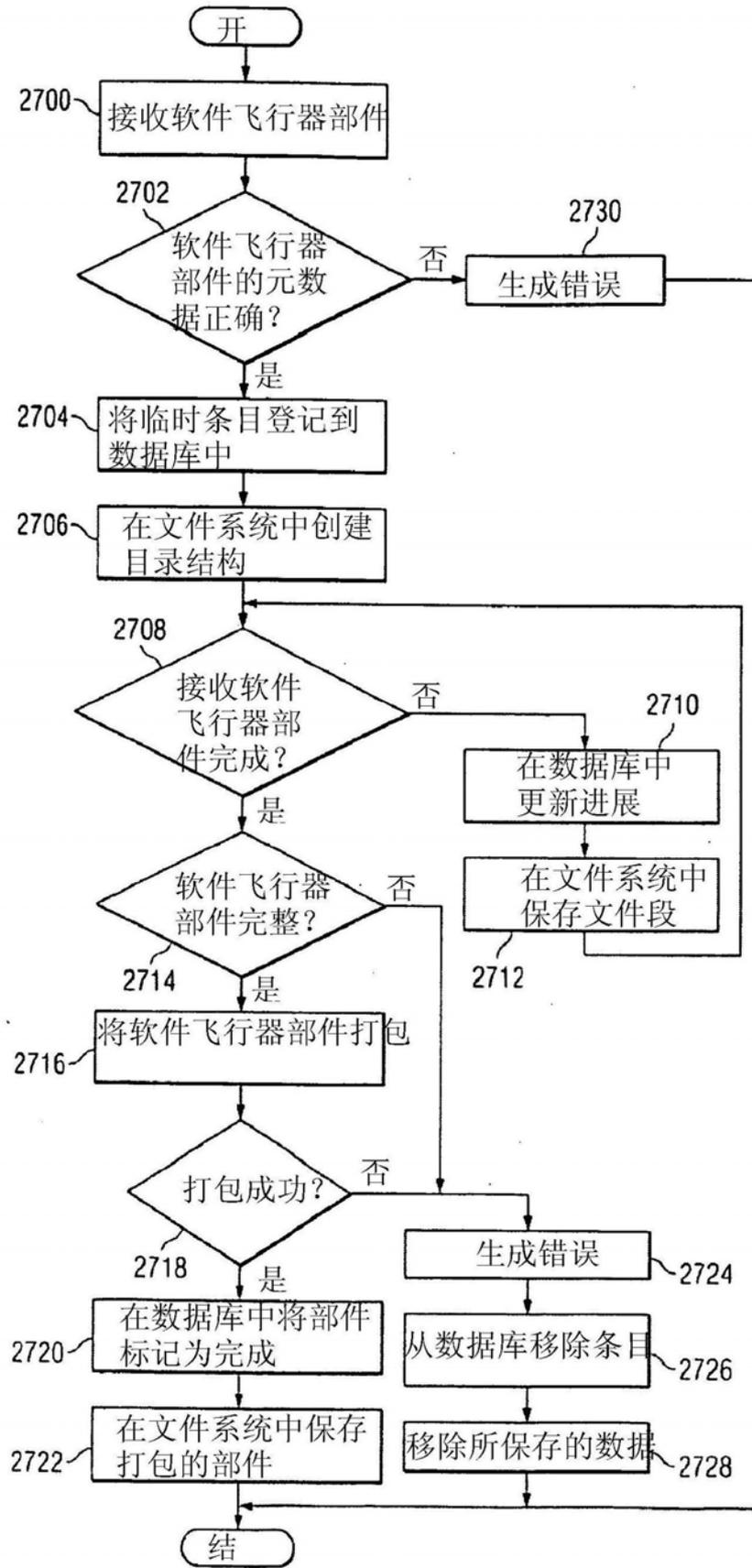


图27

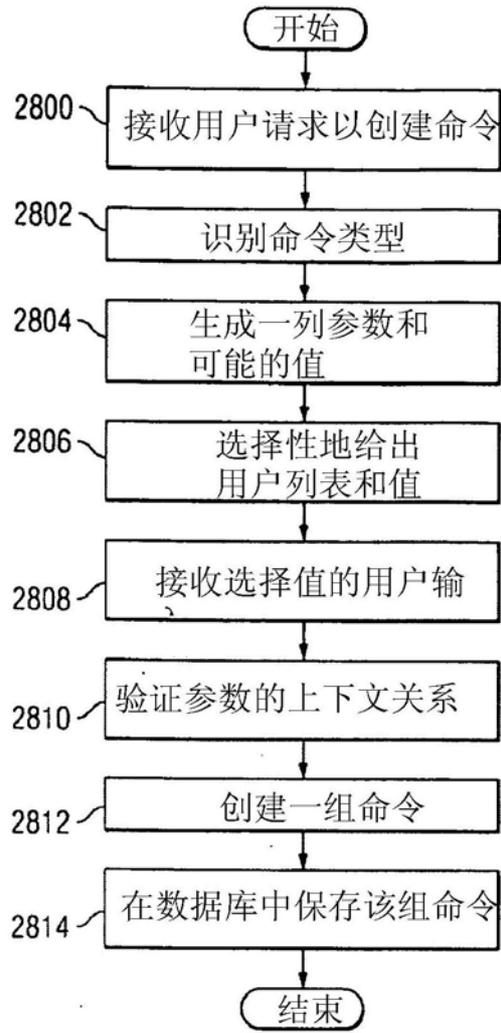


图28

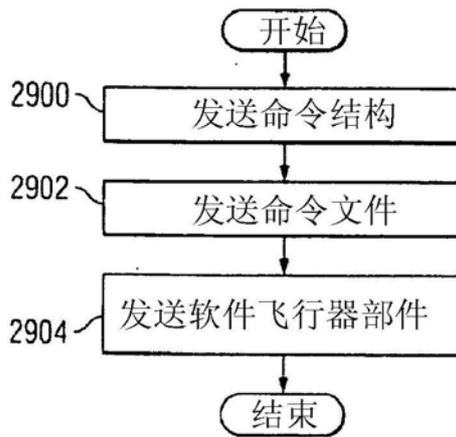


图29

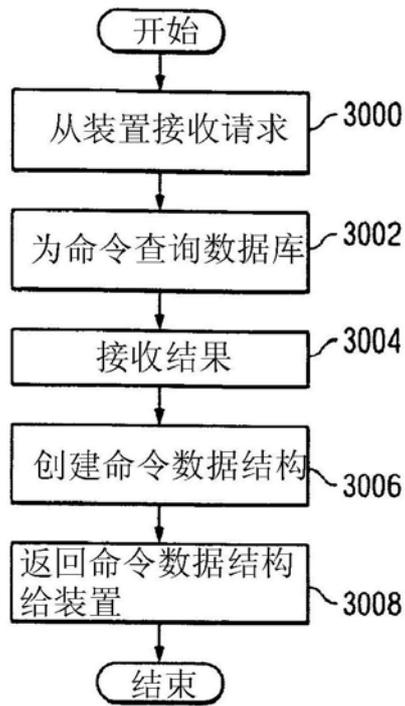


图30

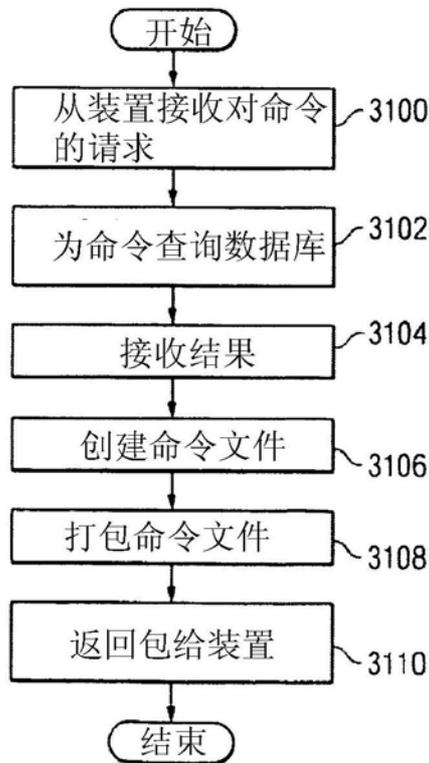


图31

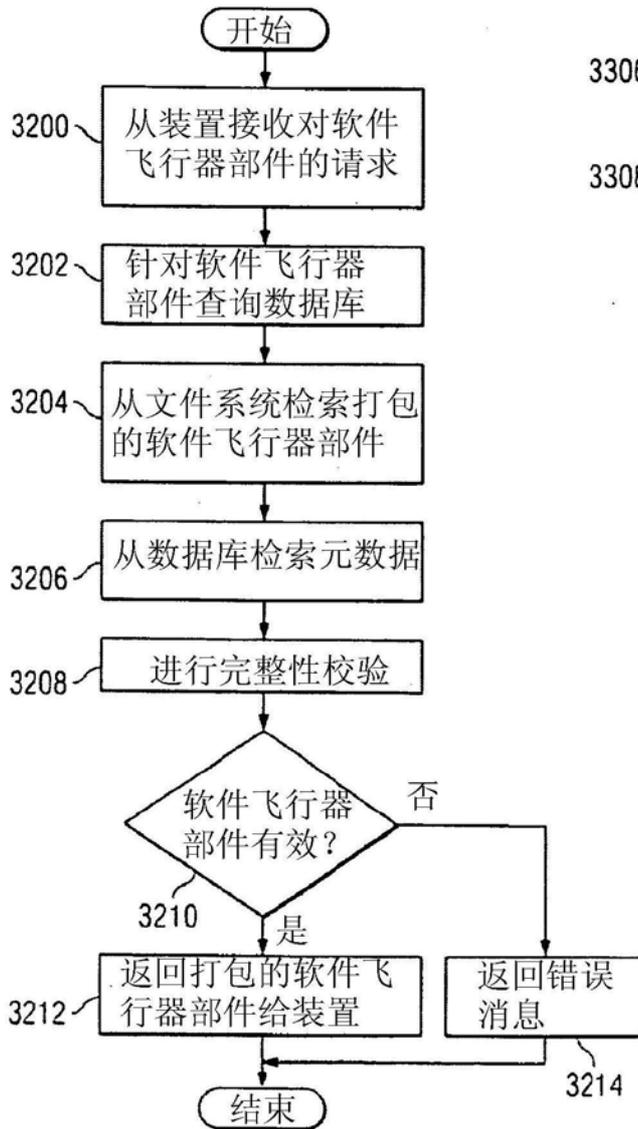


图32

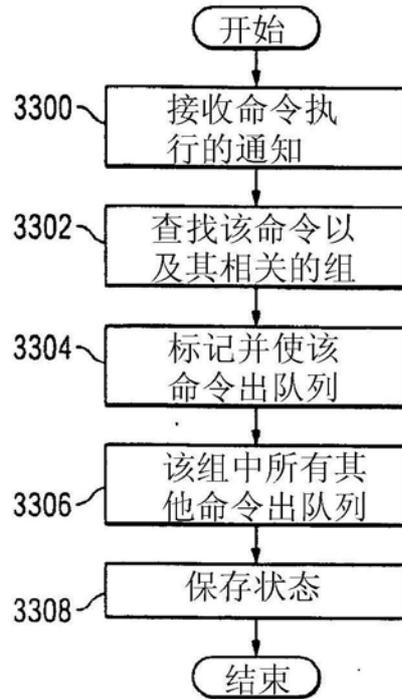


图33

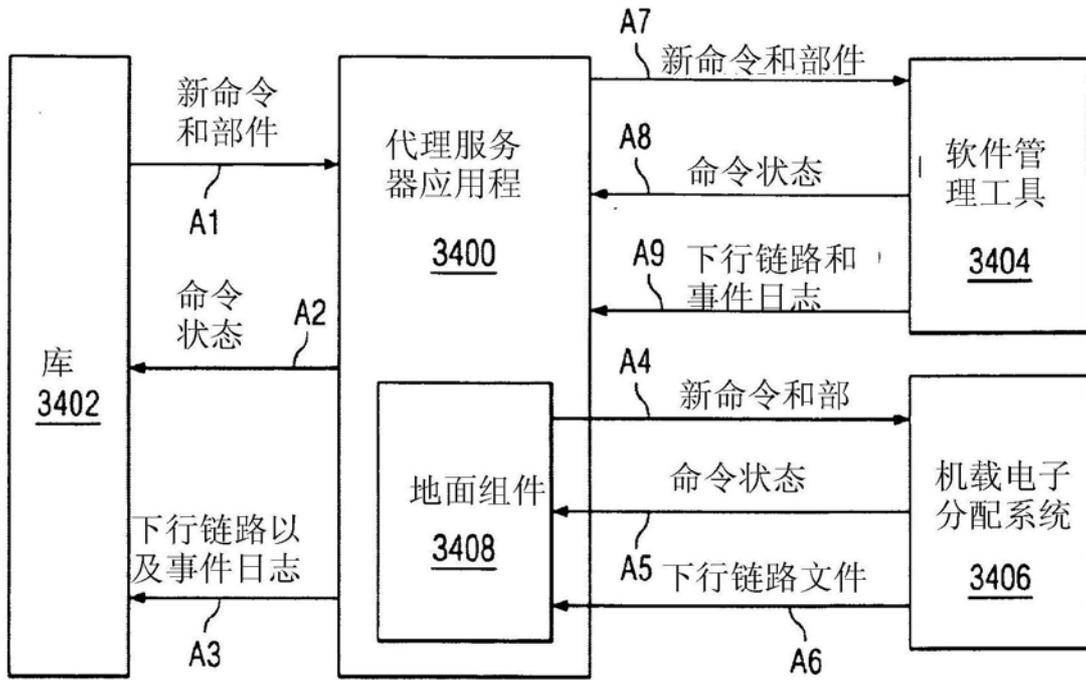


图34

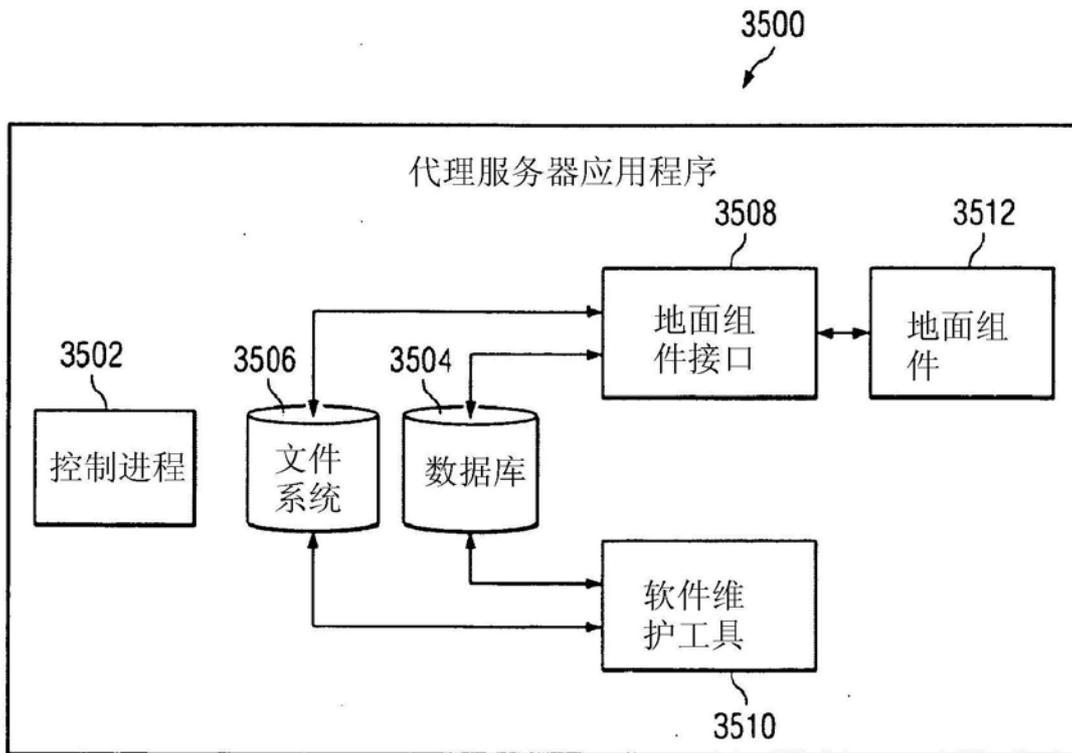


图35

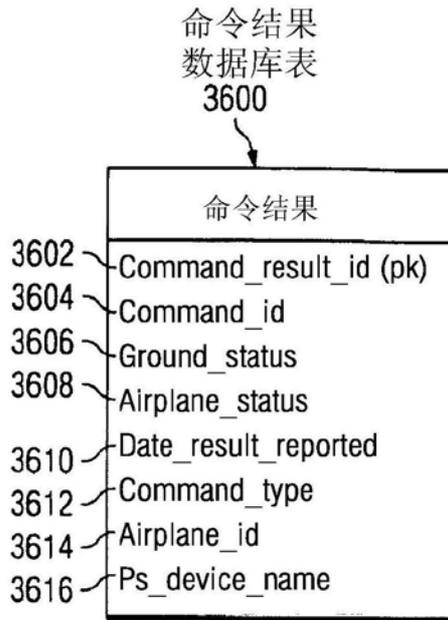


图36

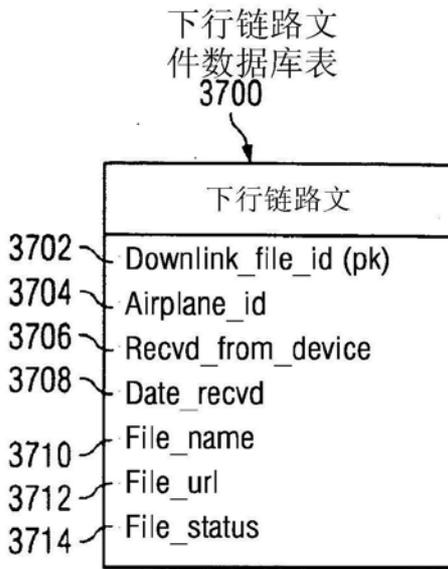


图37

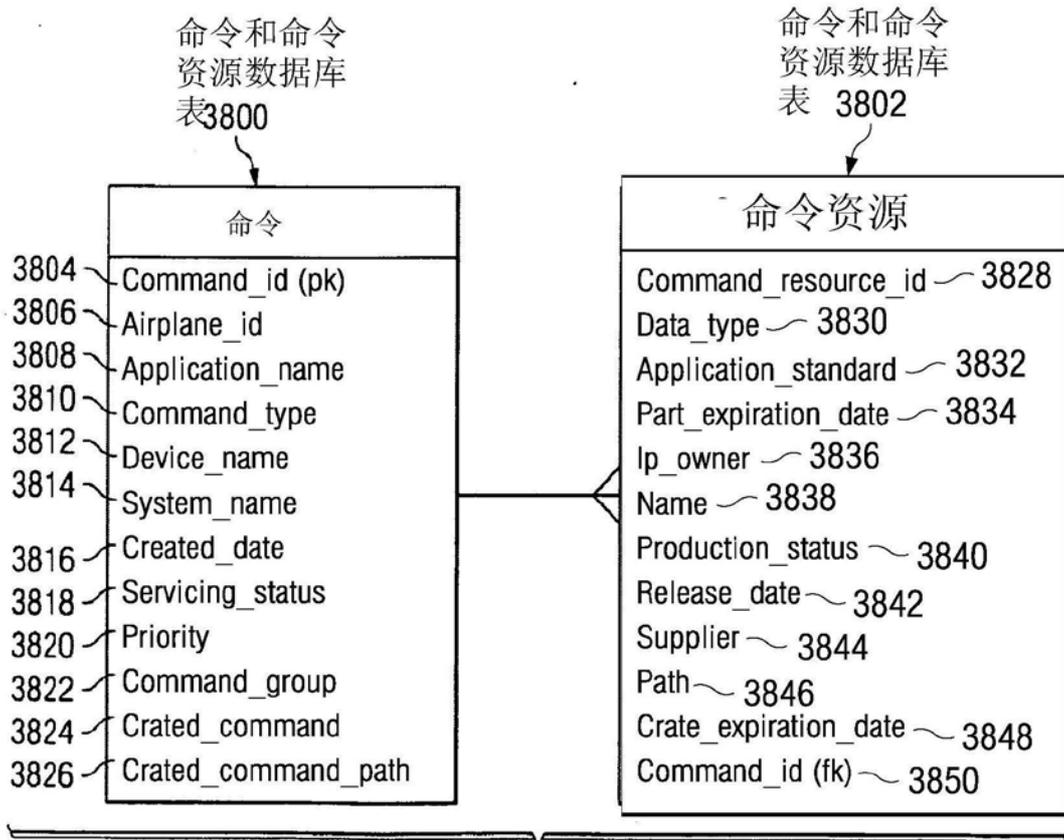


图38

图38

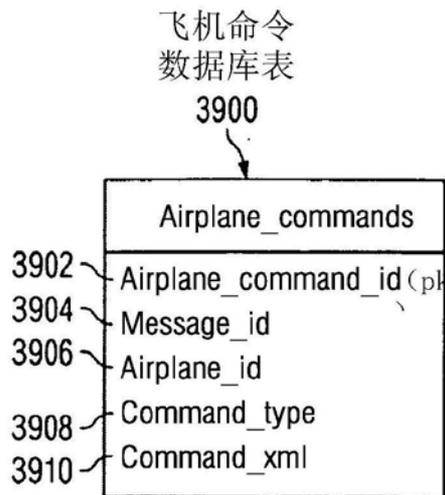


图39

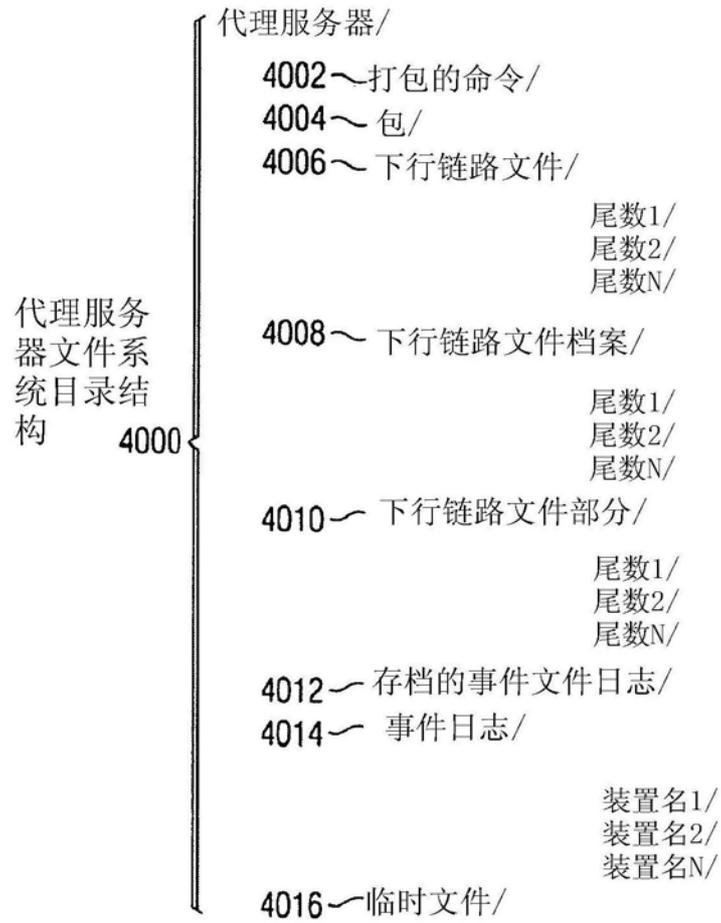


图40

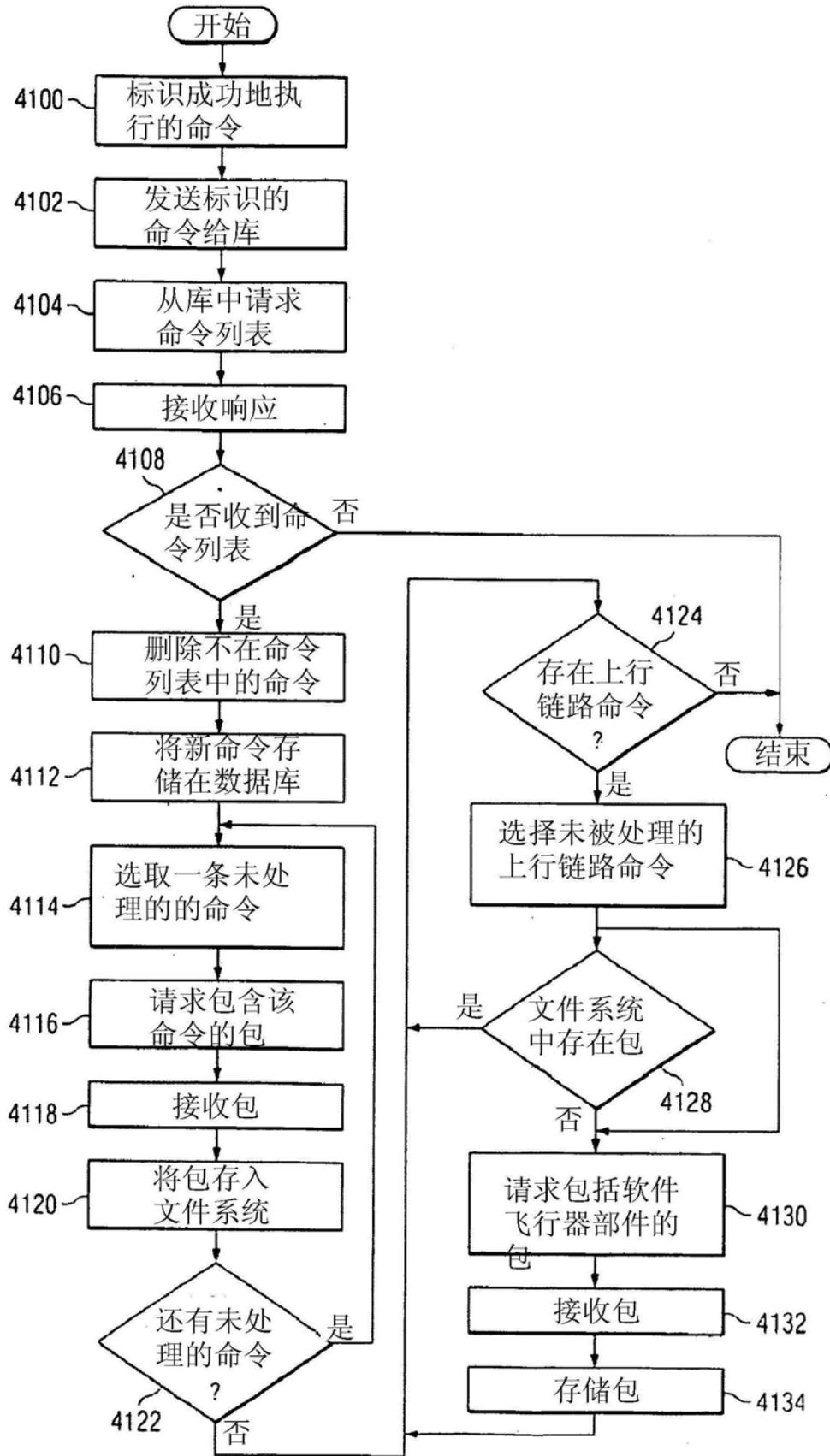


图41

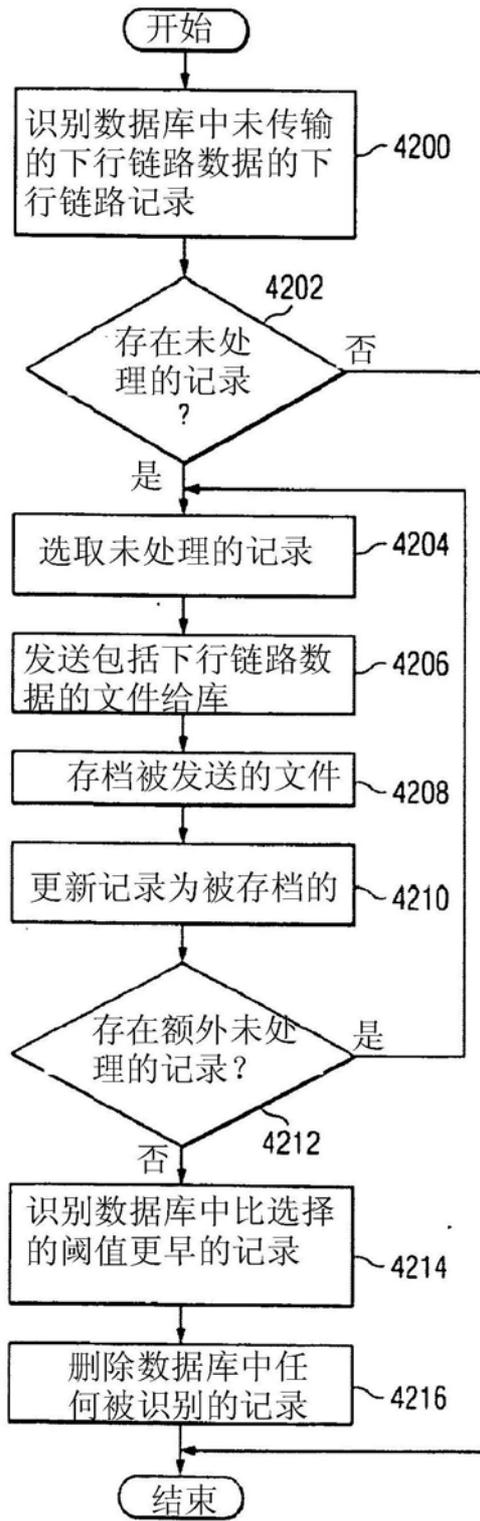


图42

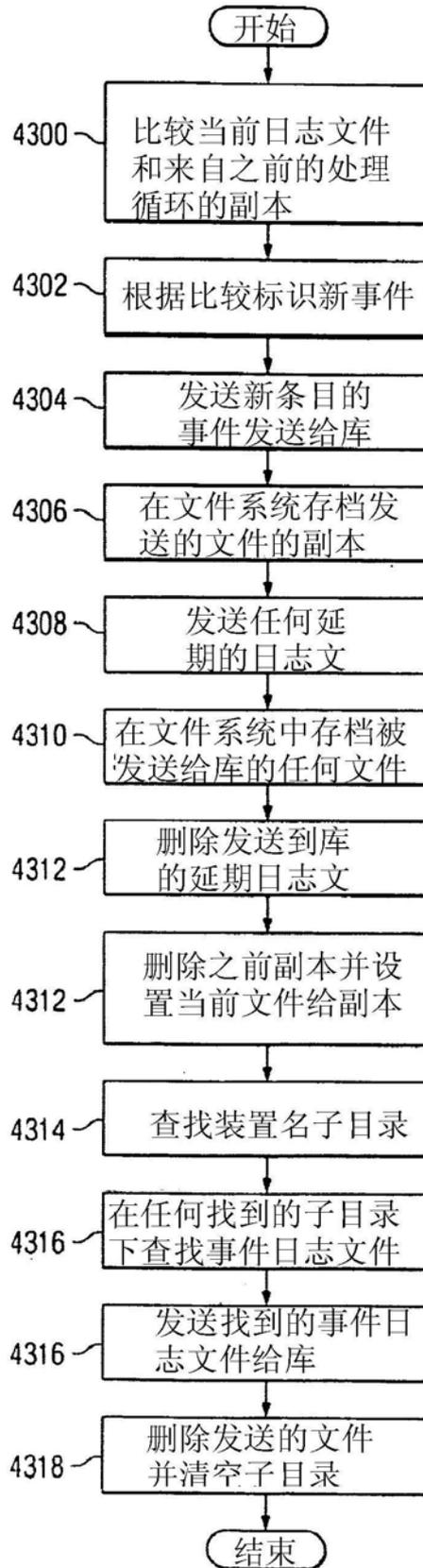


图43

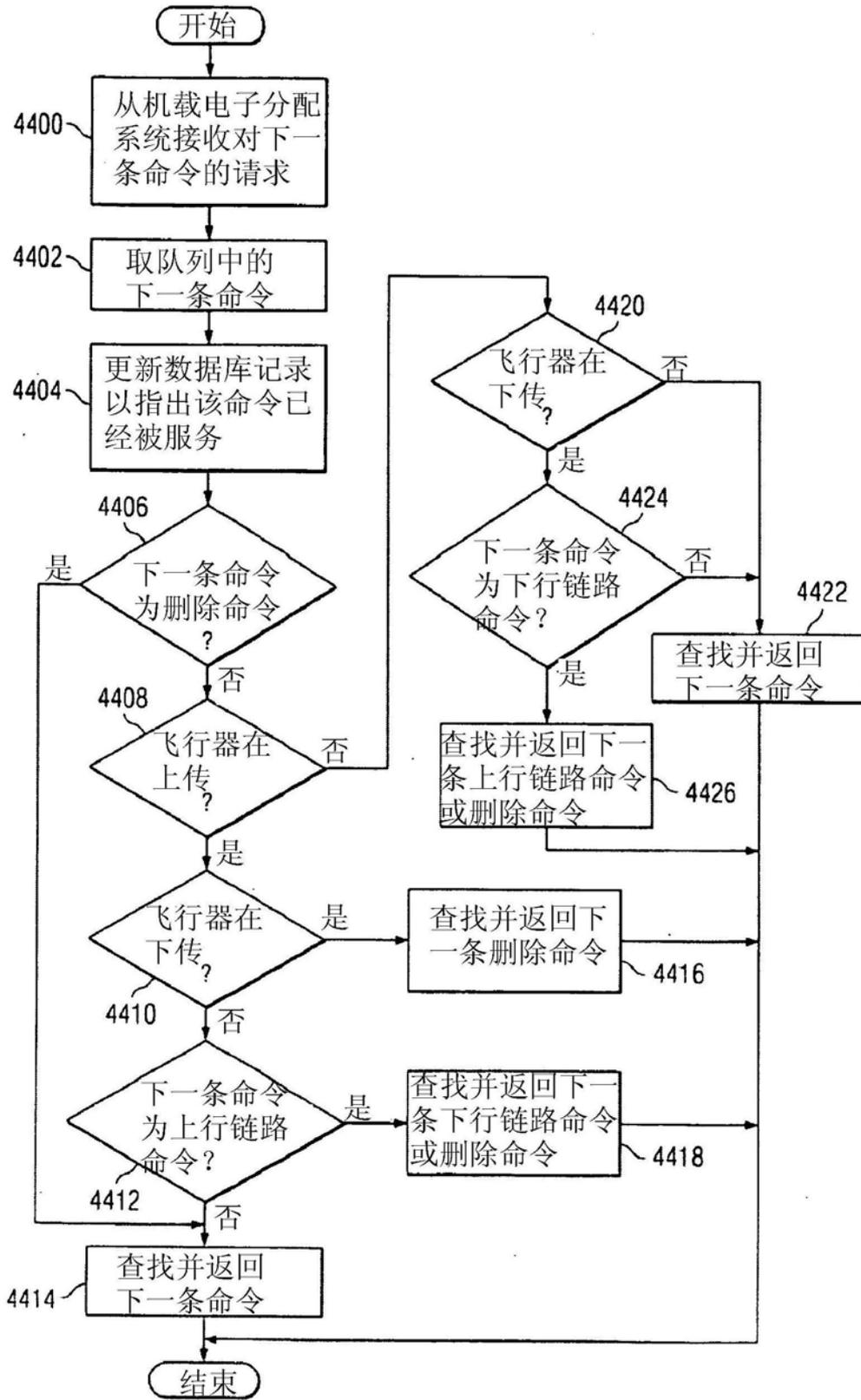


图44

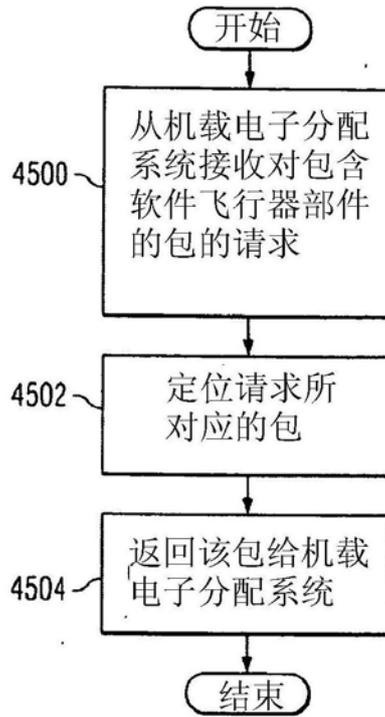


图45

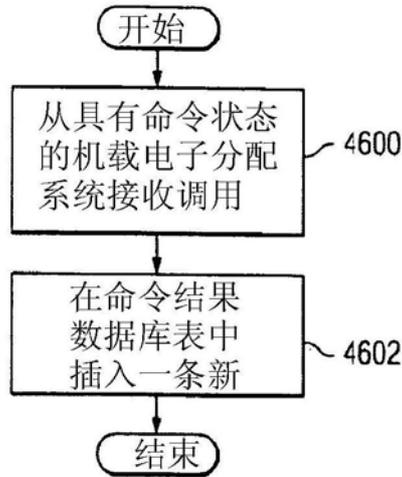


图46

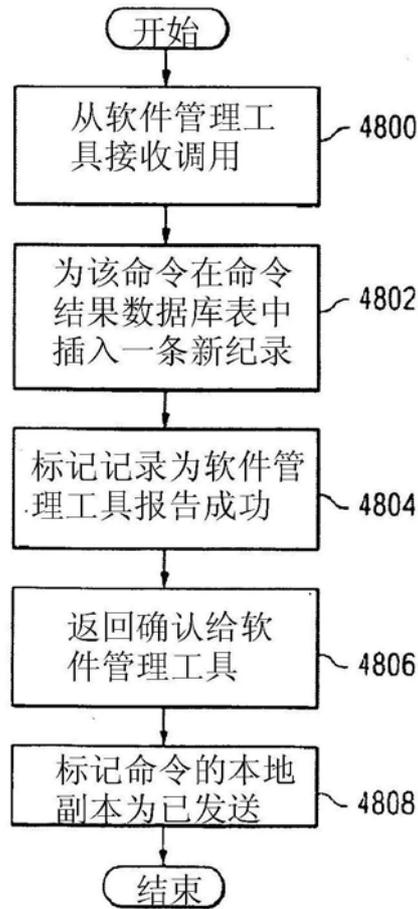


图48

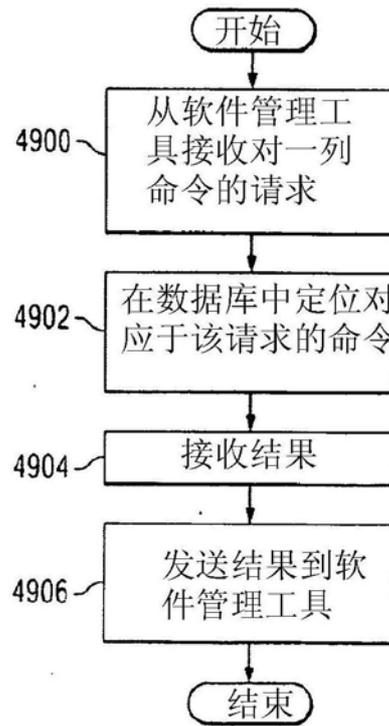


图49

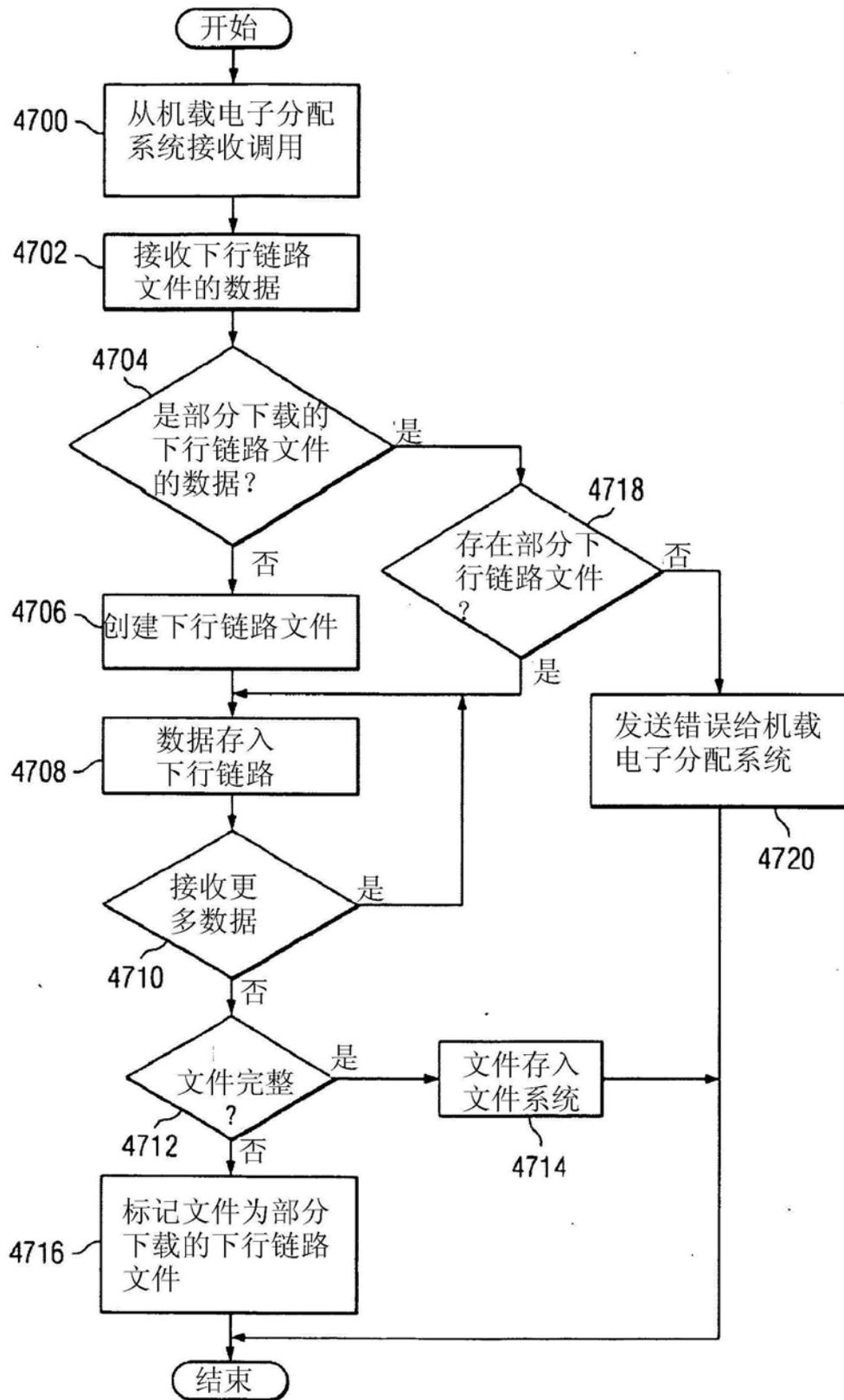


图47

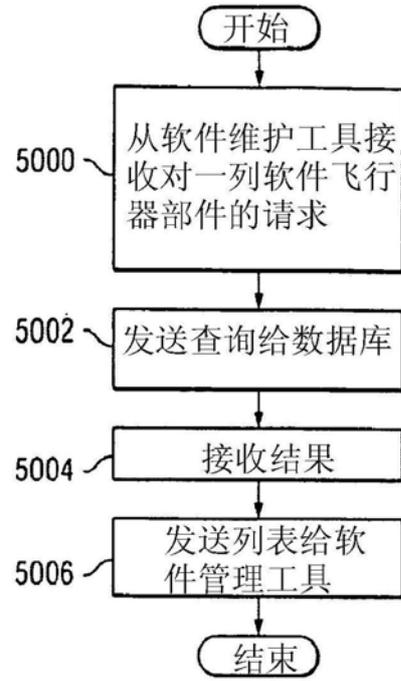


图50

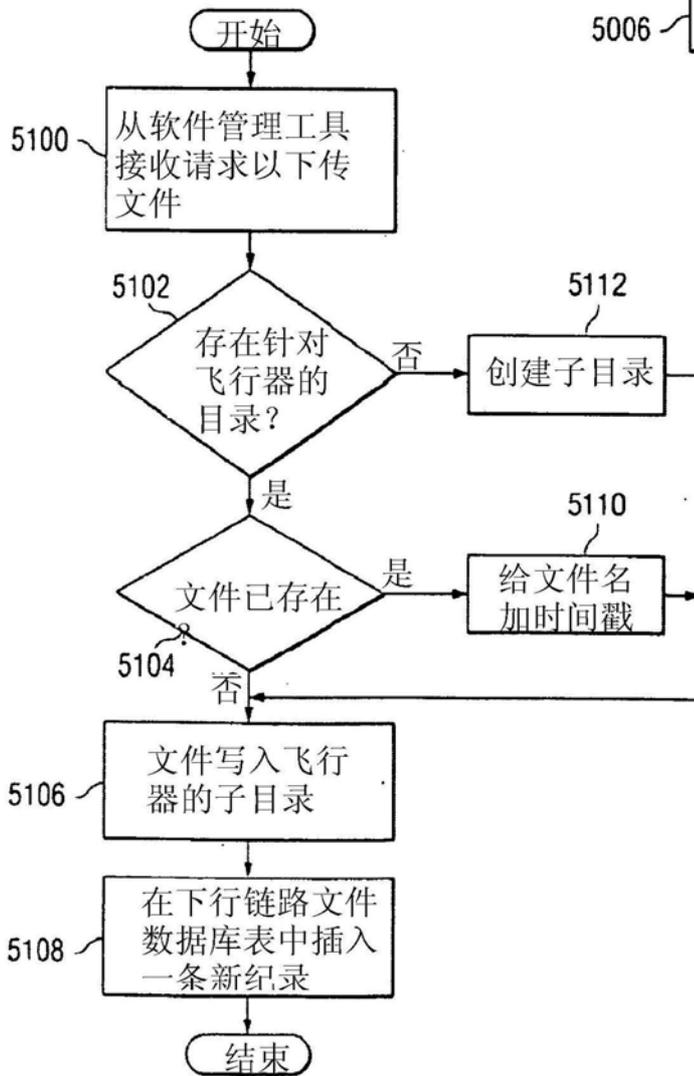


图51

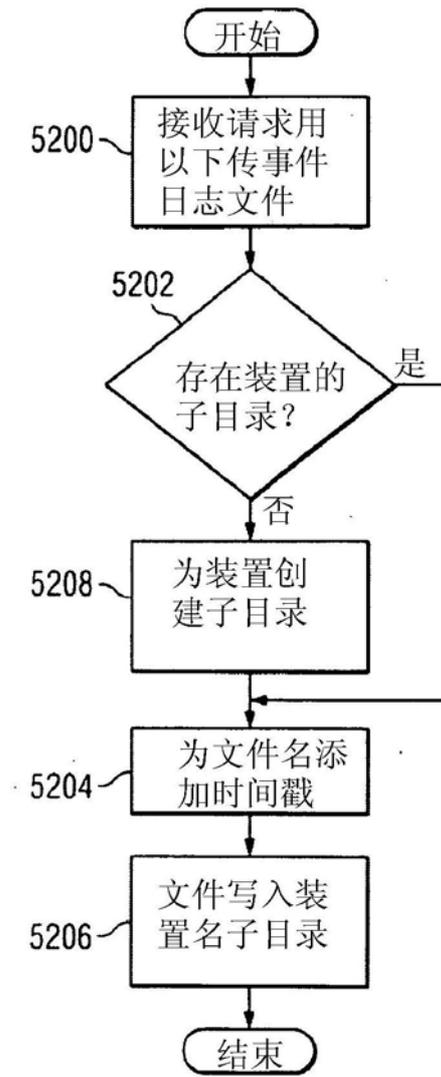


图52

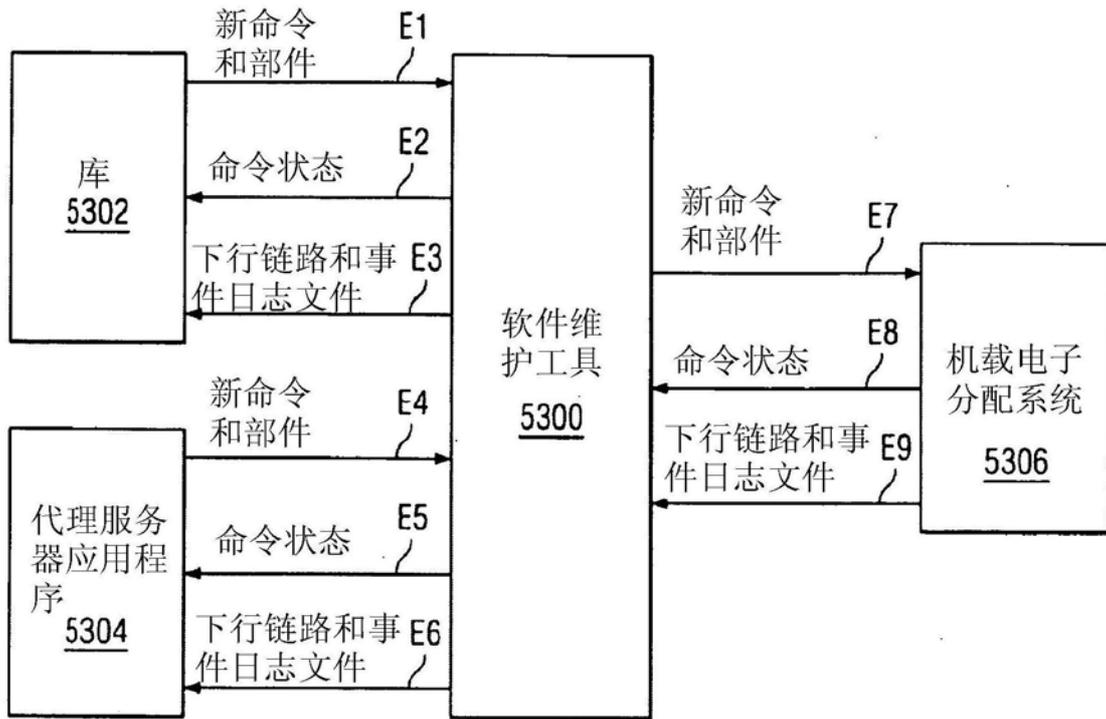


图53

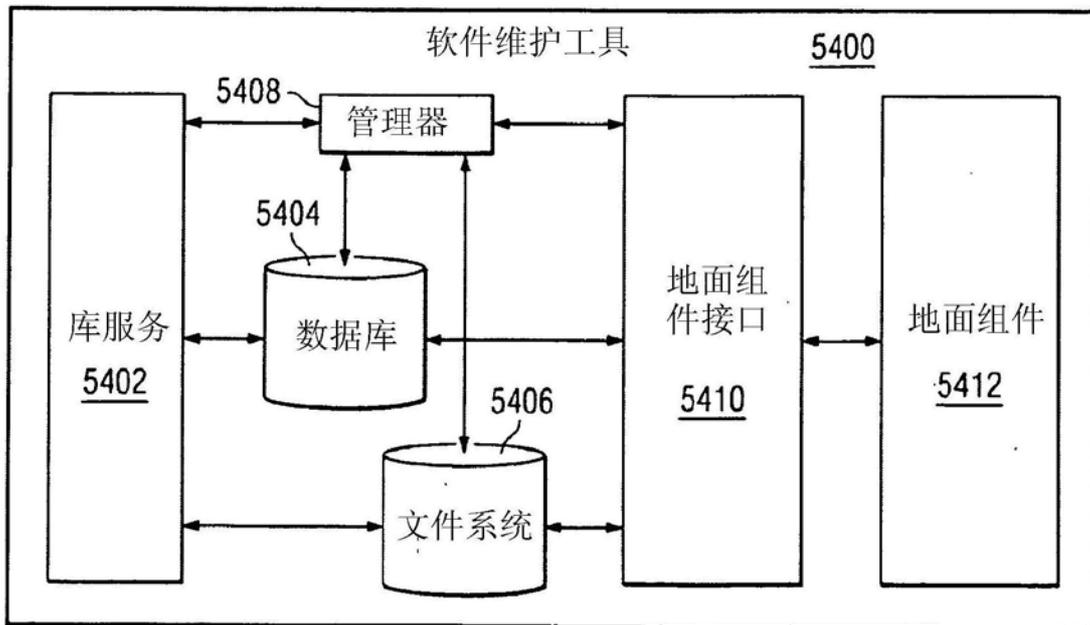


图54

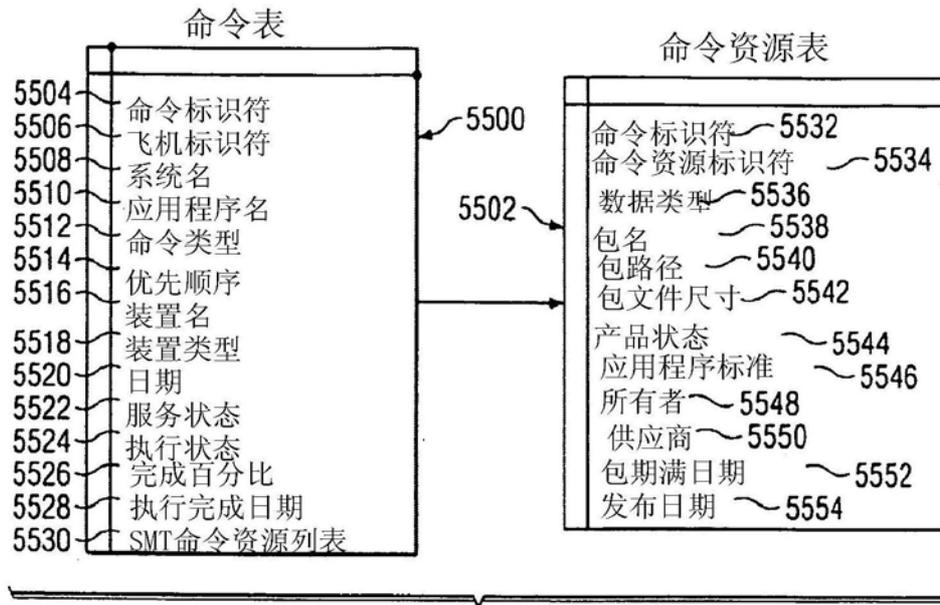


图55

图55

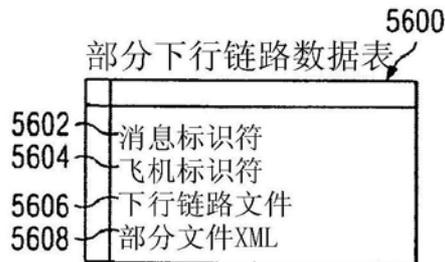


图56

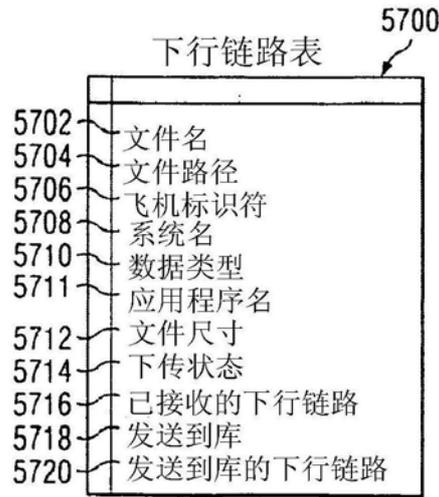


图57

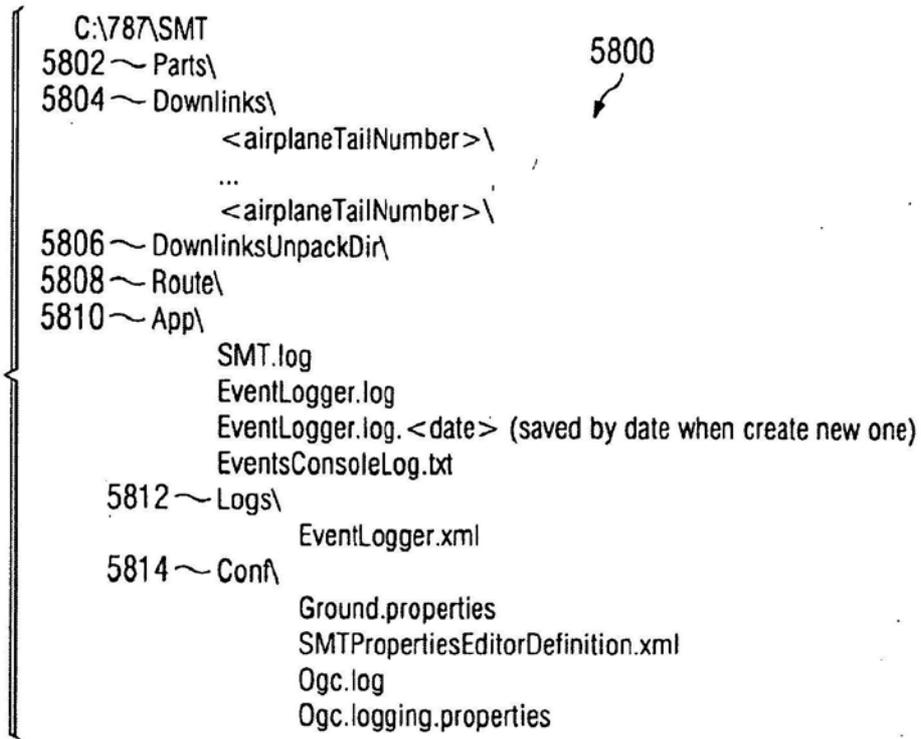


图58

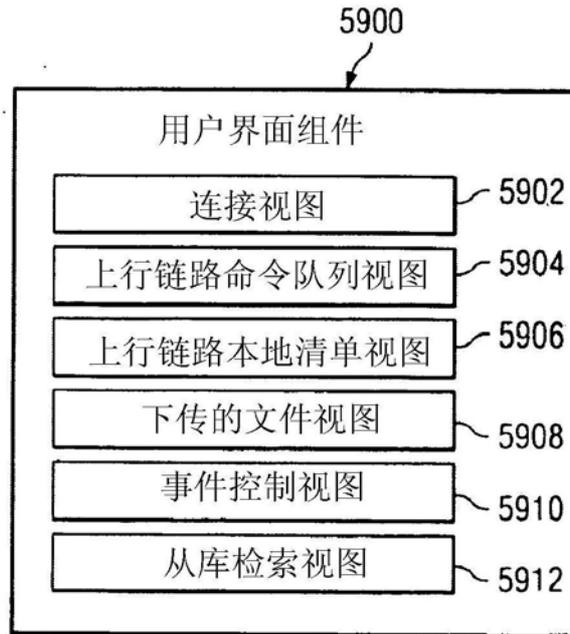


图59

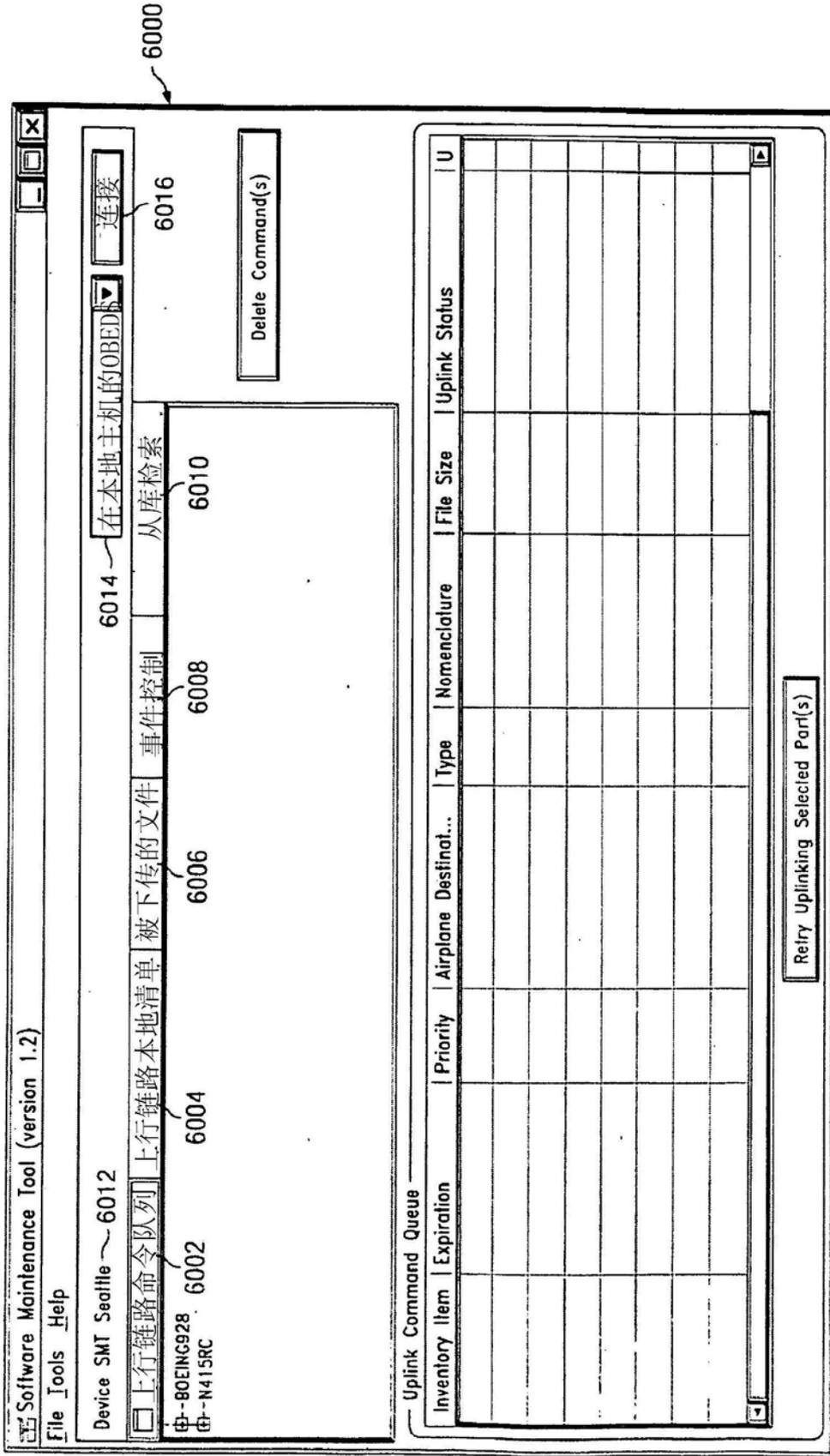


图60

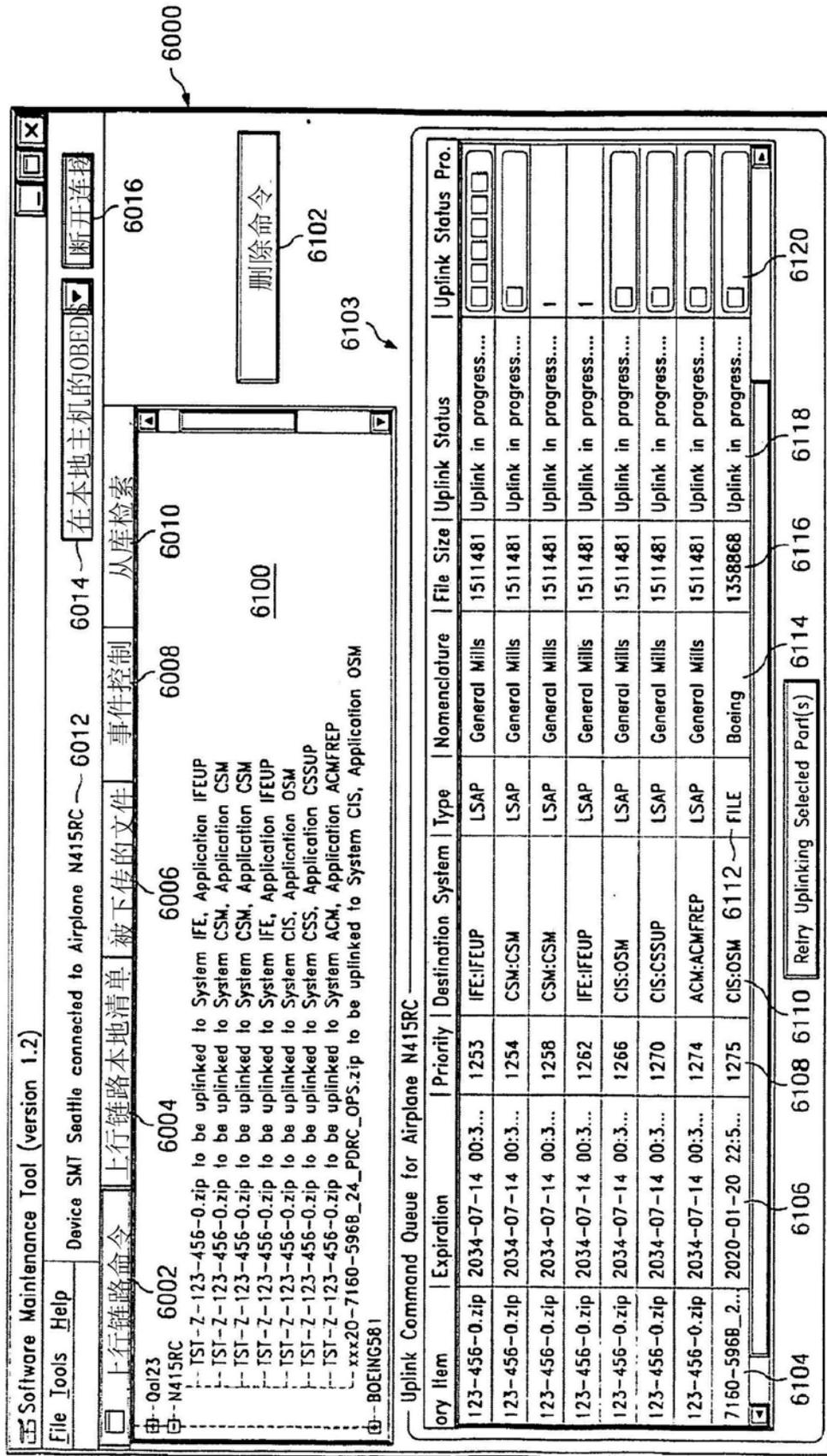


图61

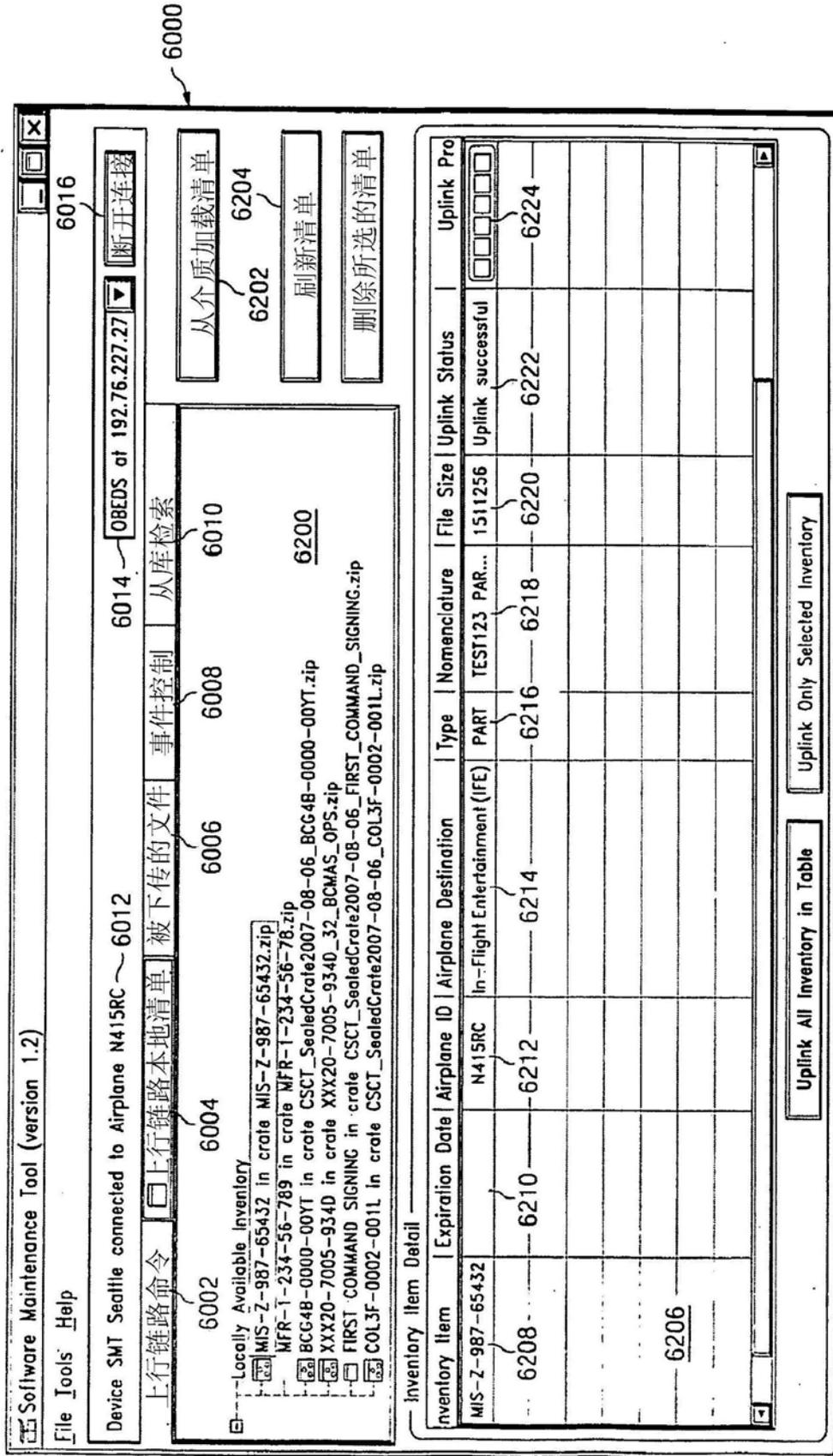


图62

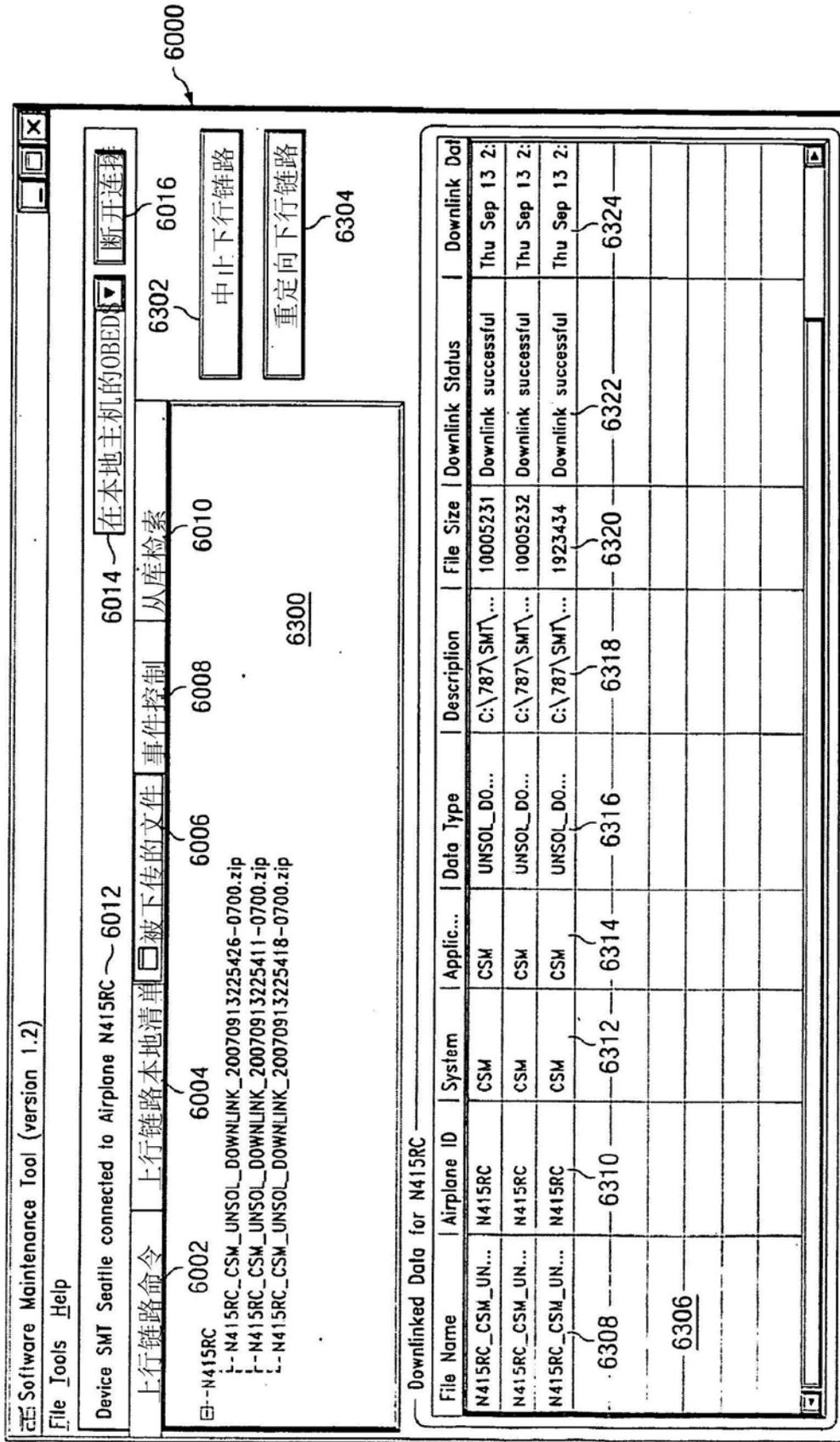


图63

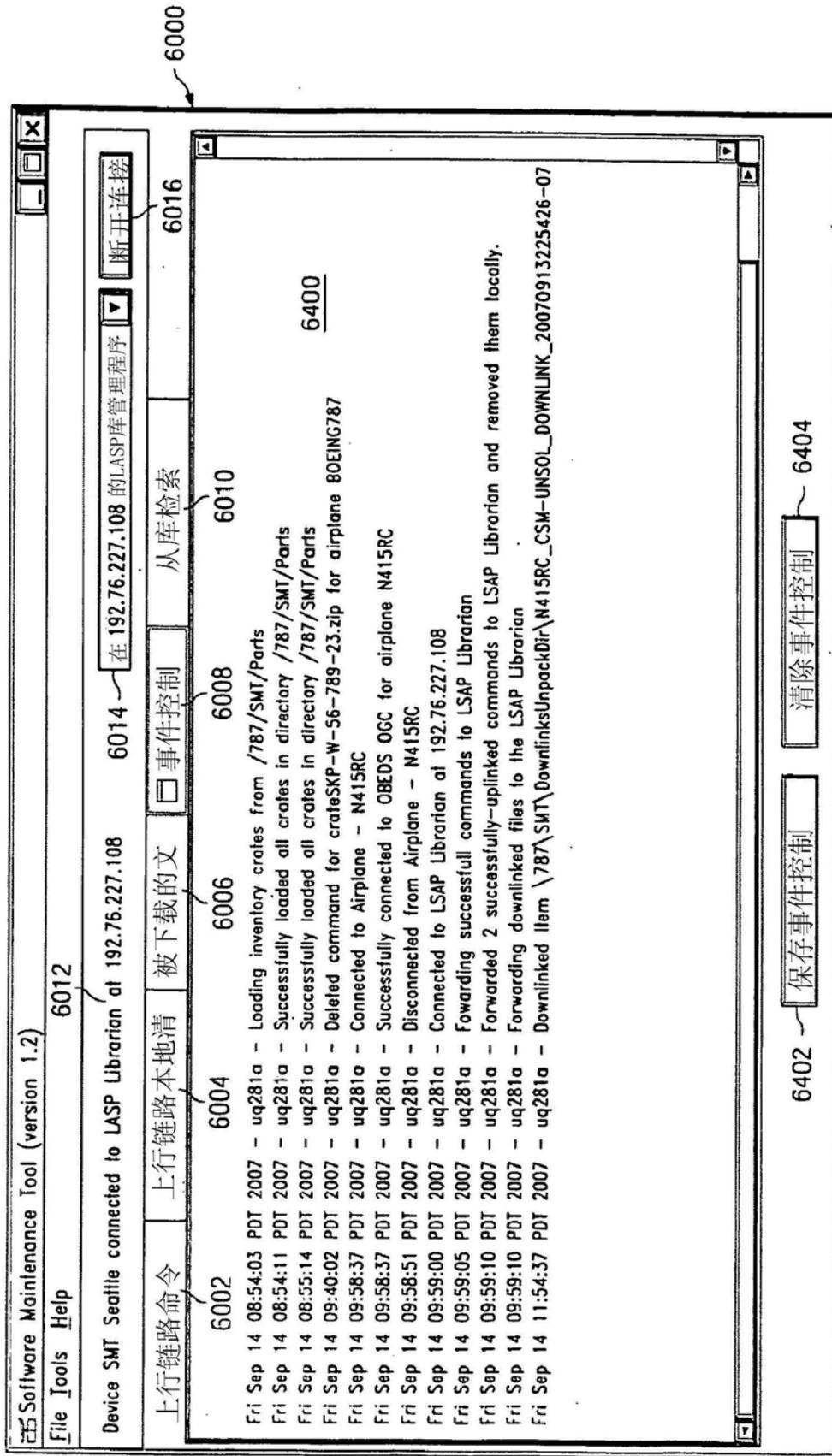


图64

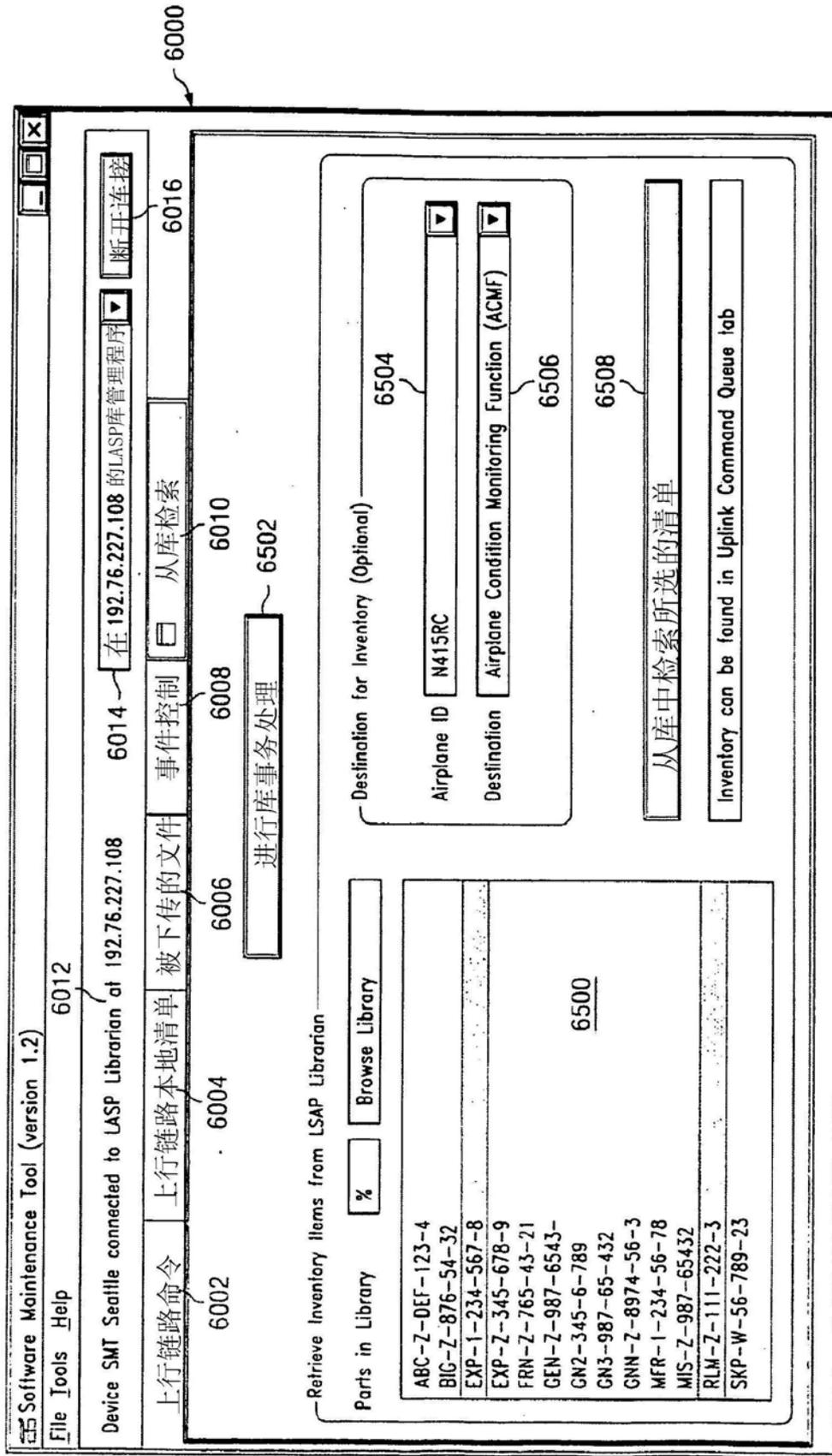


图65

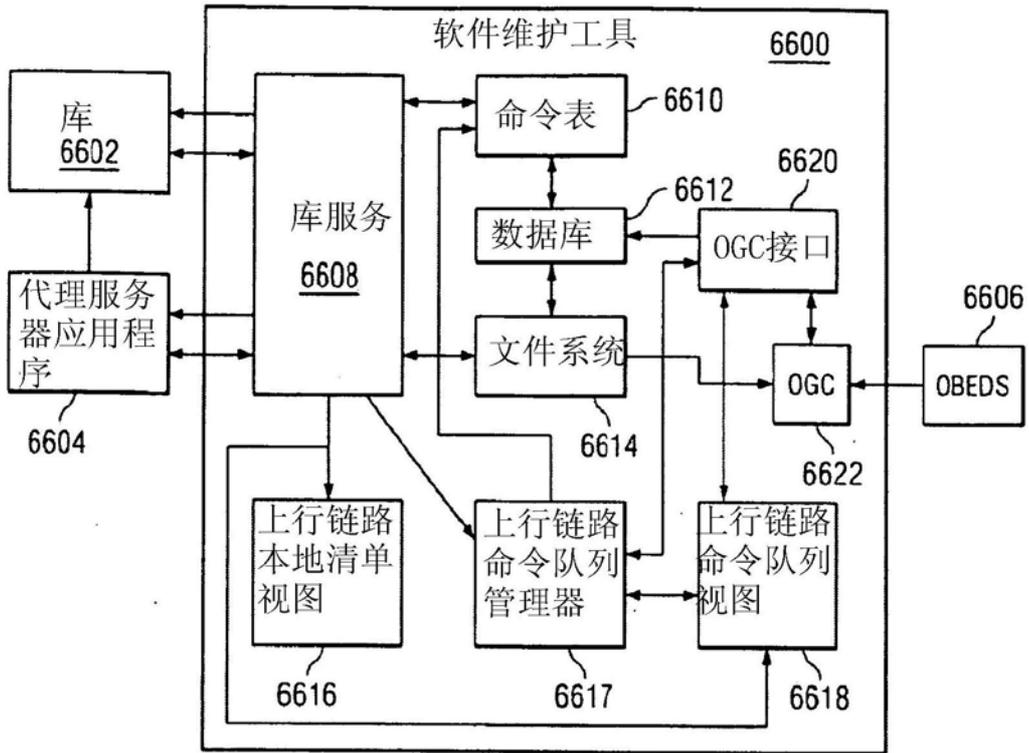


图66

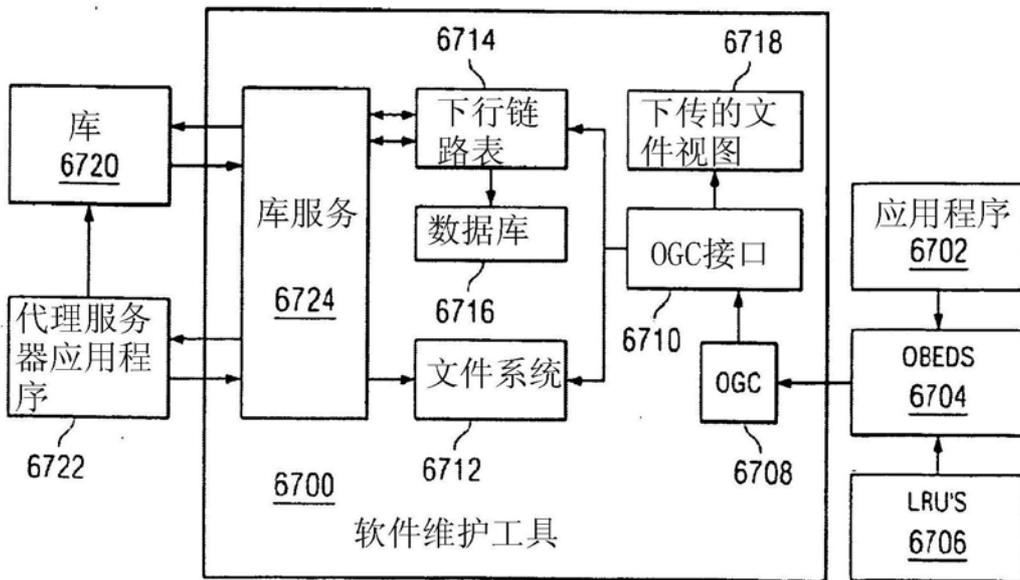


图67

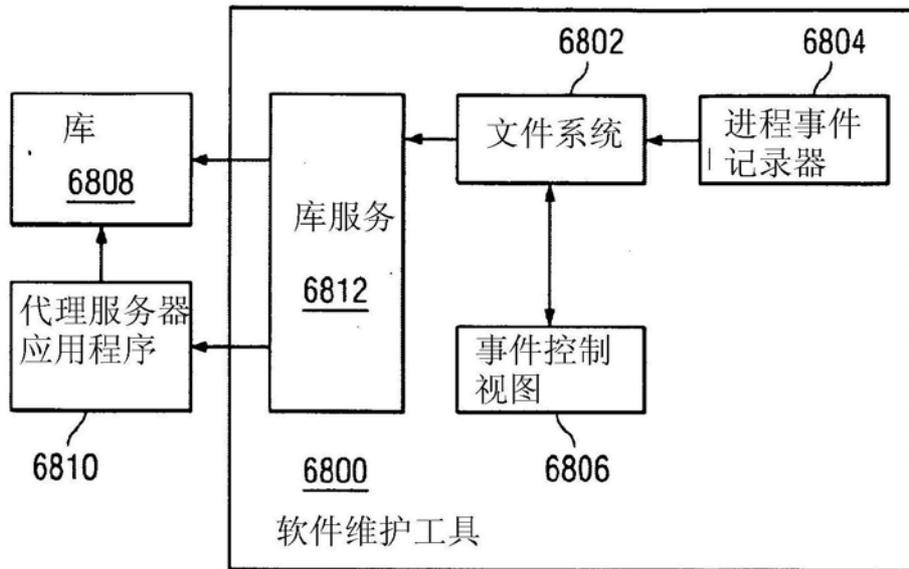


图68

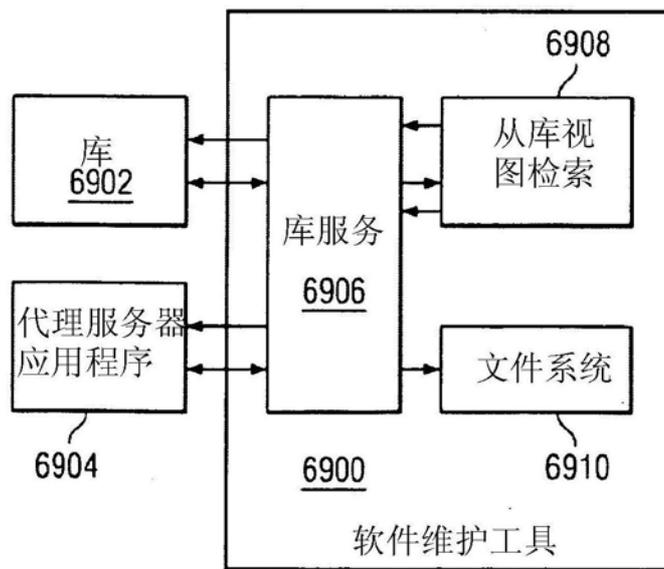


图69

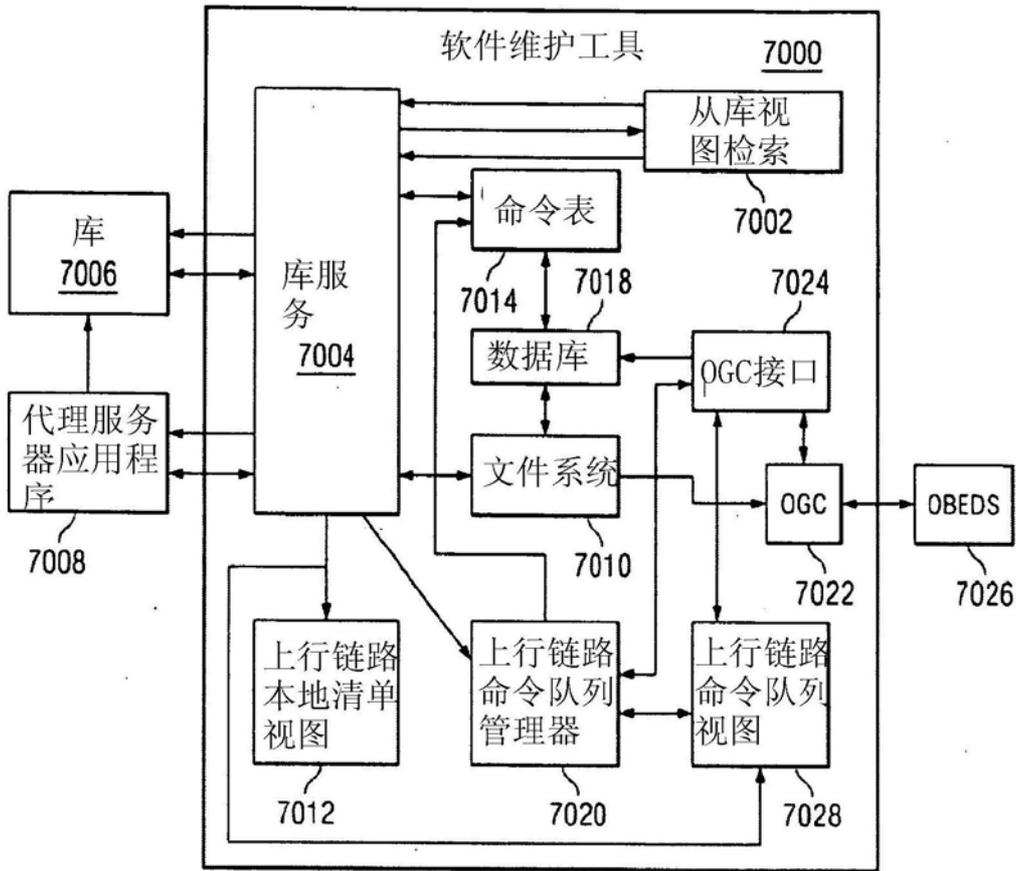


图70

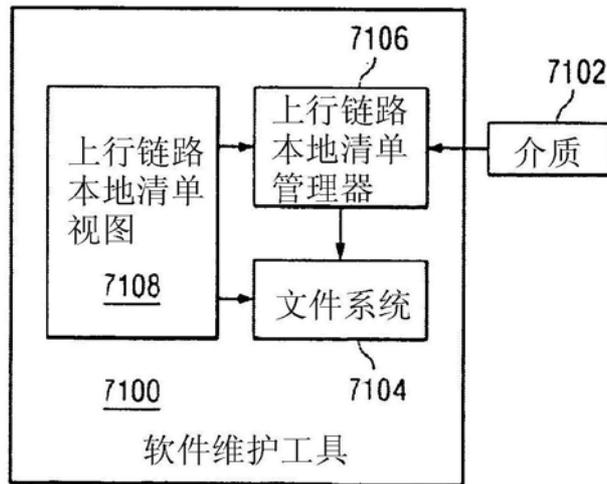


图71

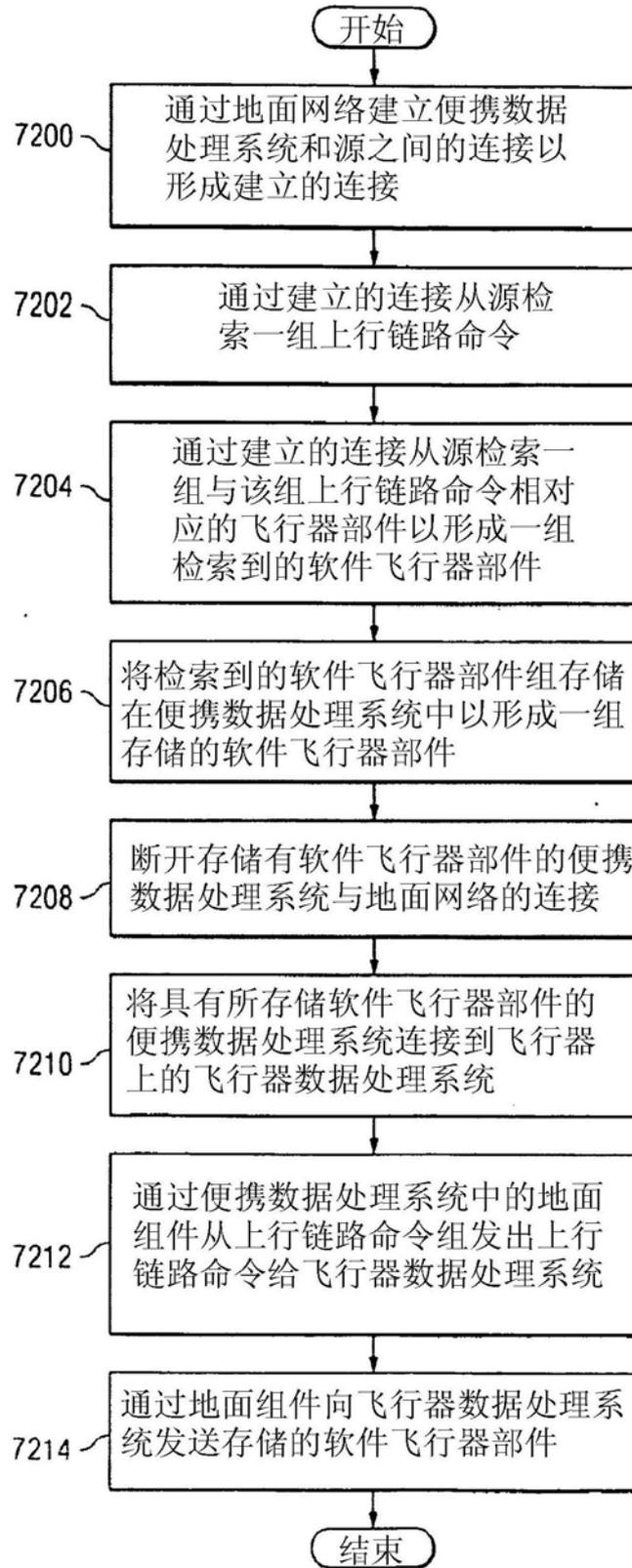


图72

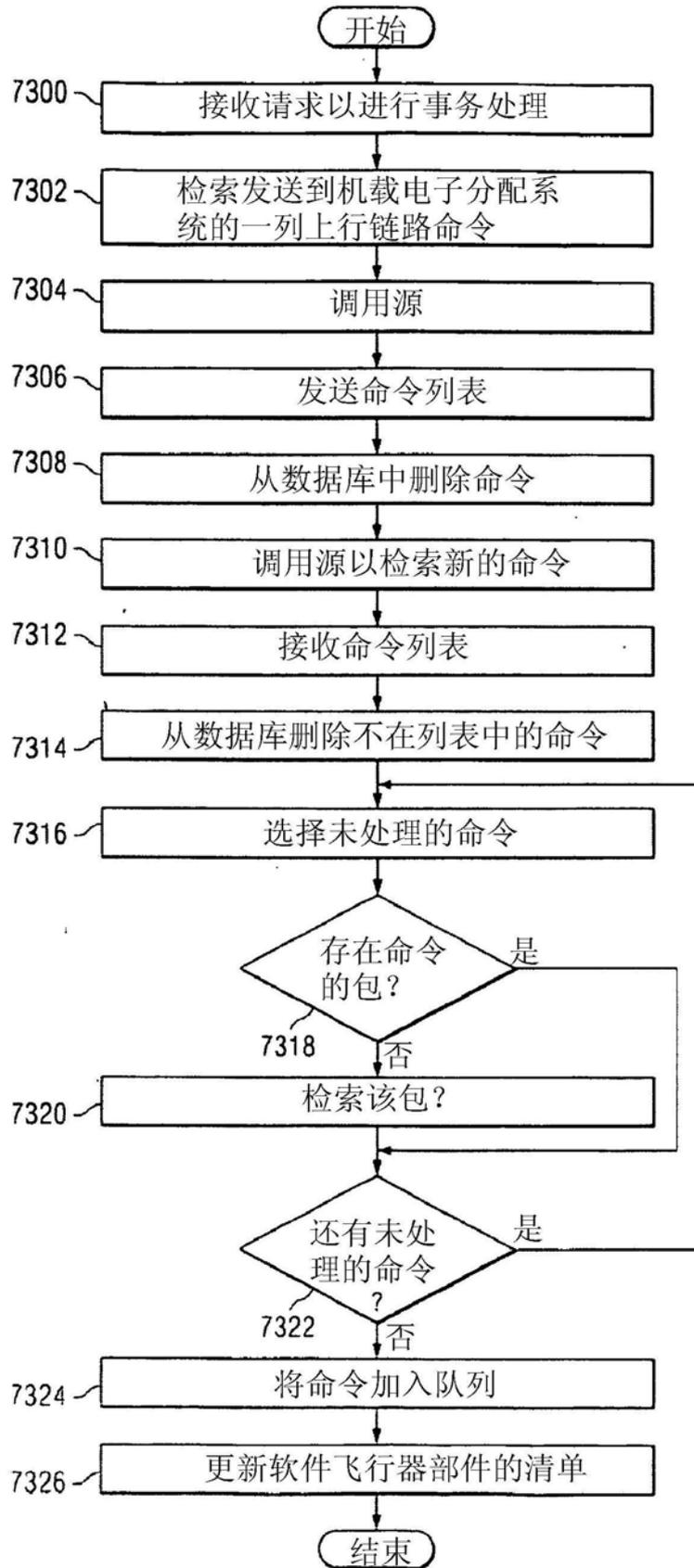


图73

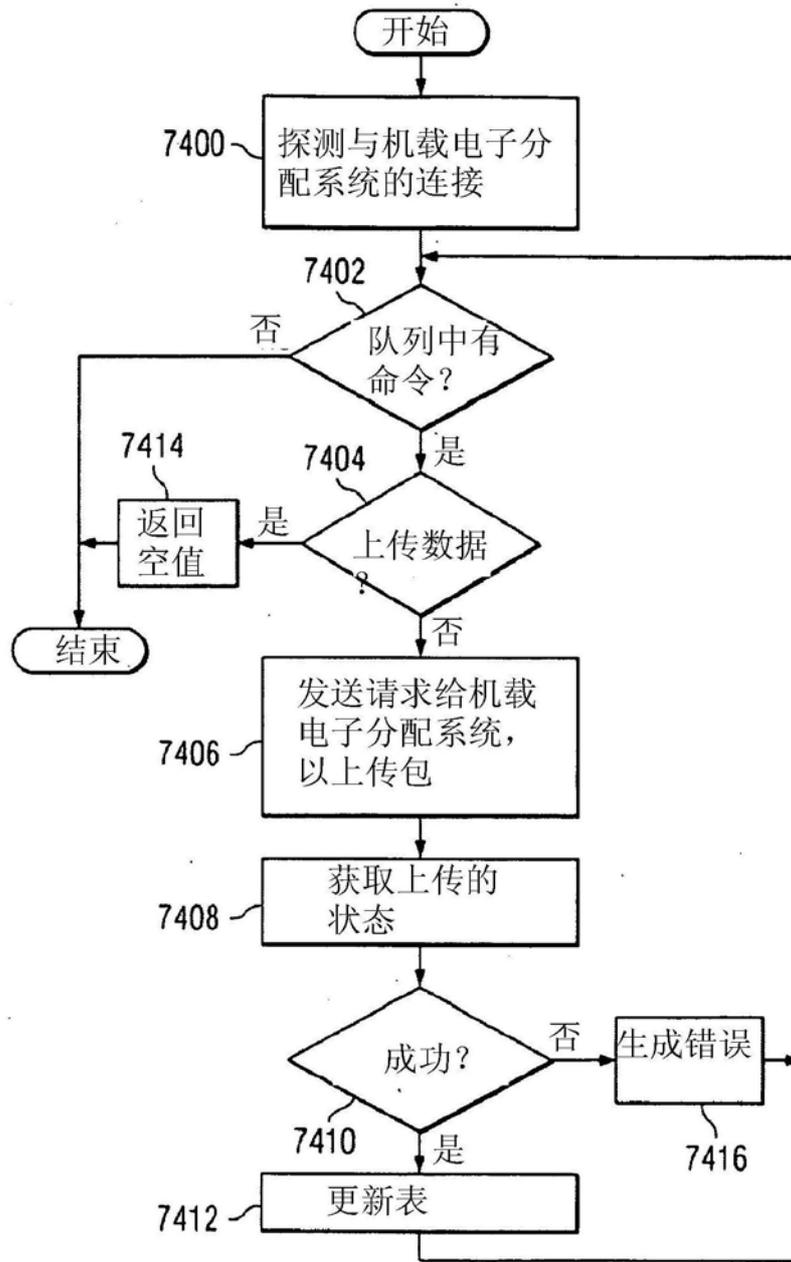


图74

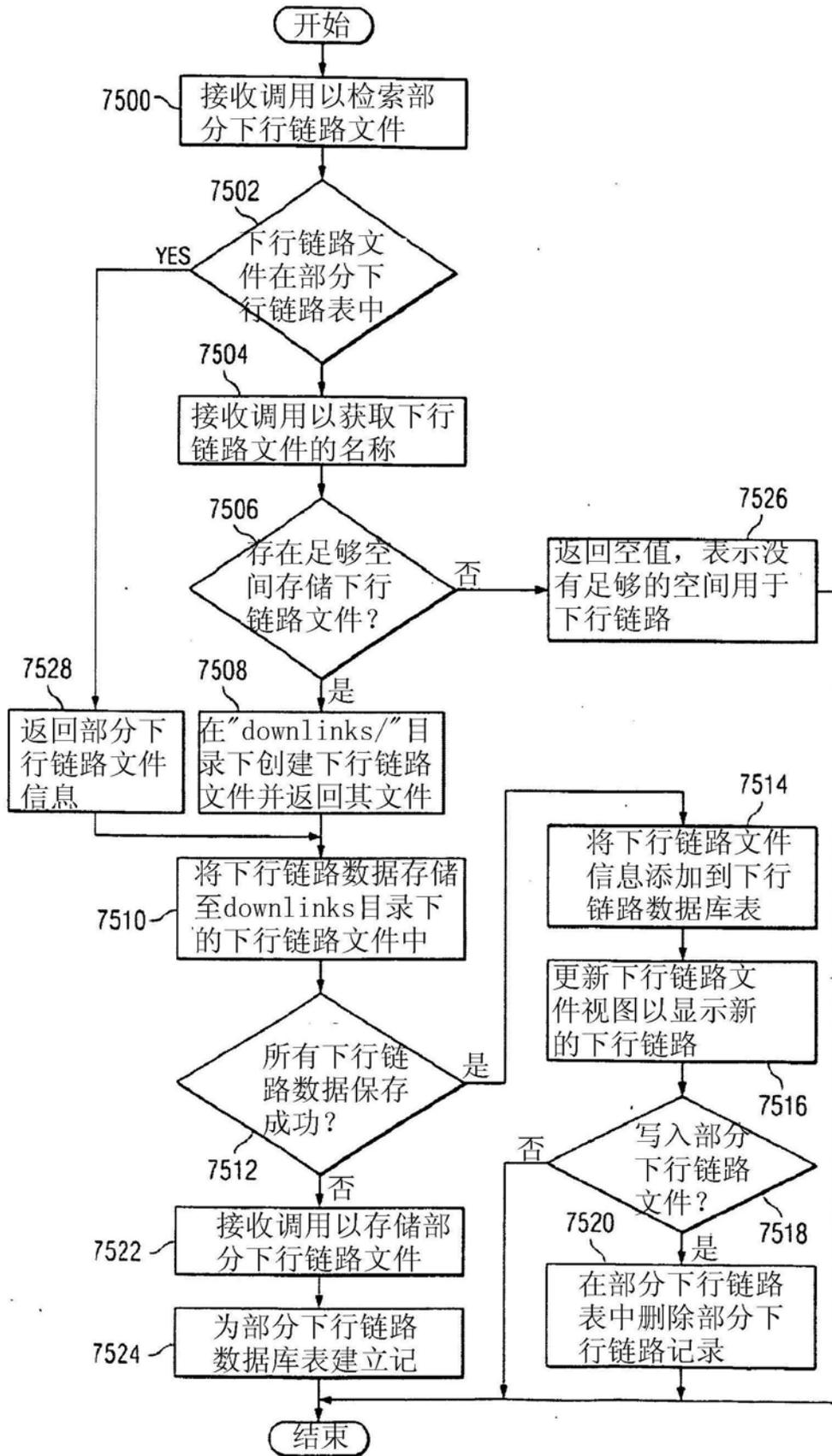


图75

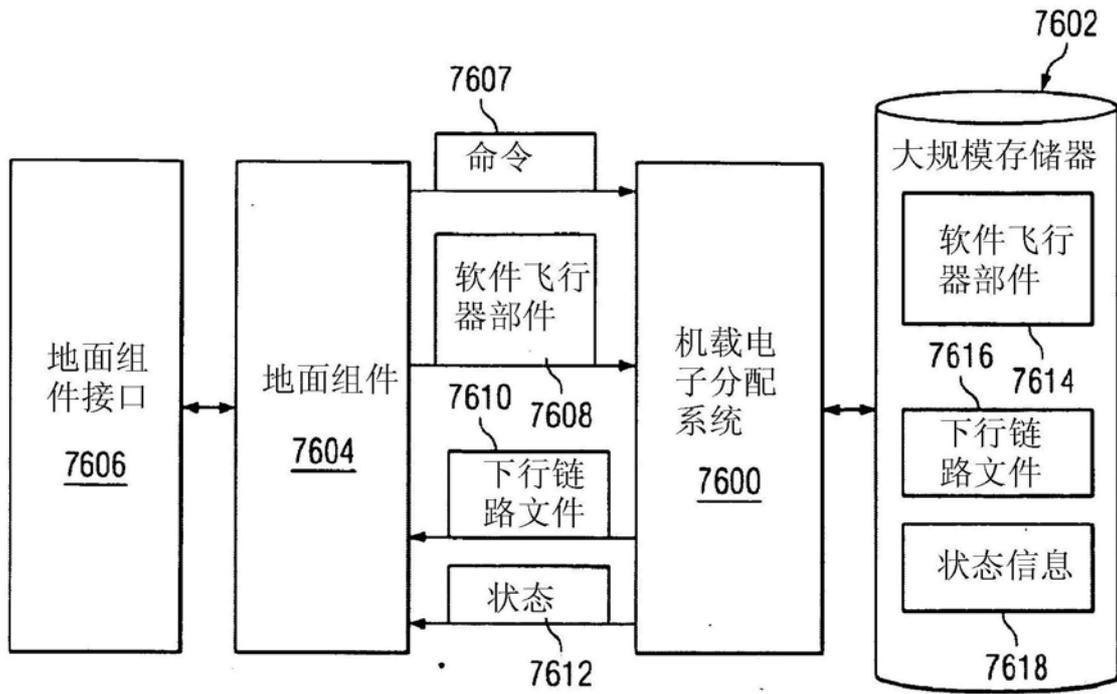


图76

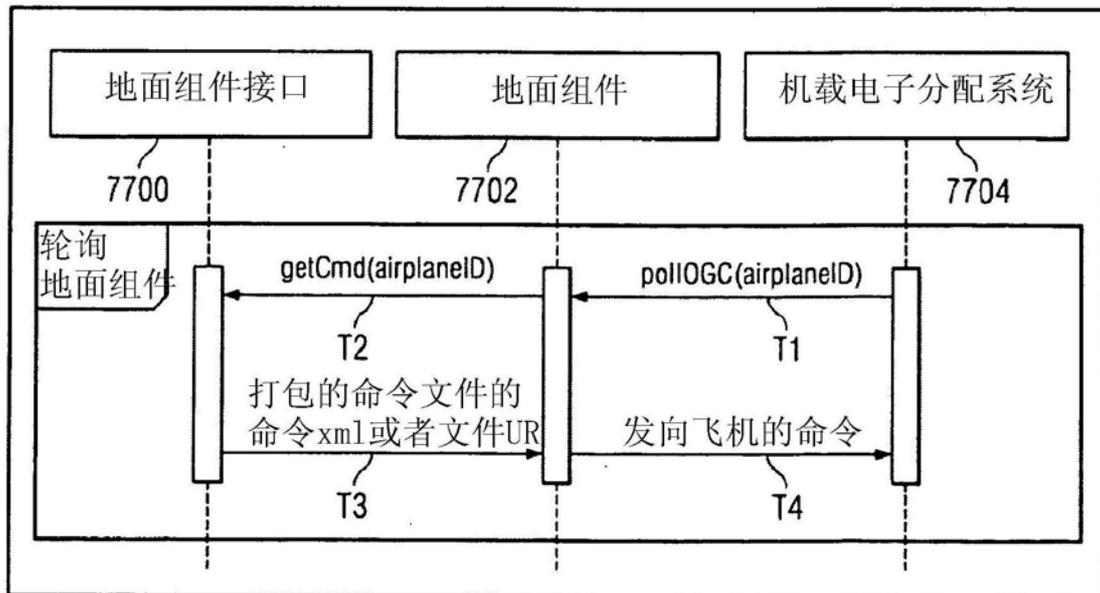


图77

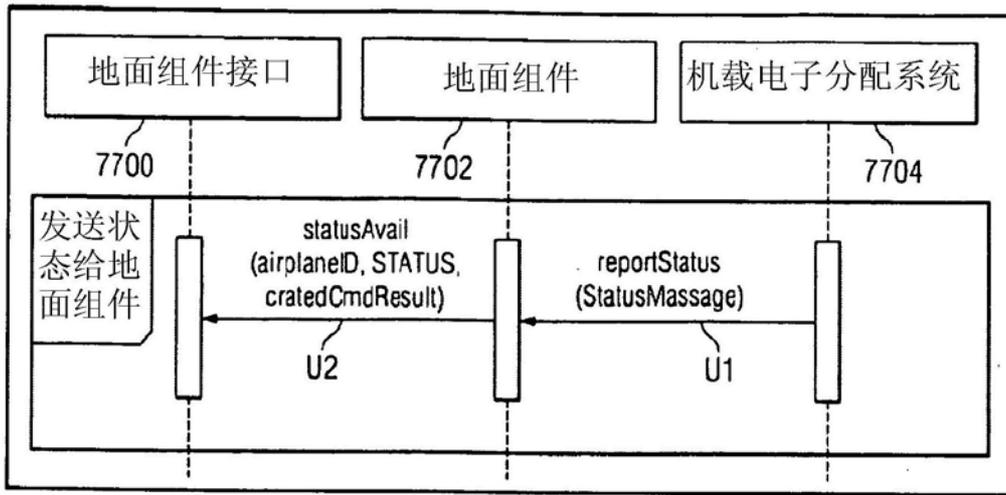


图78

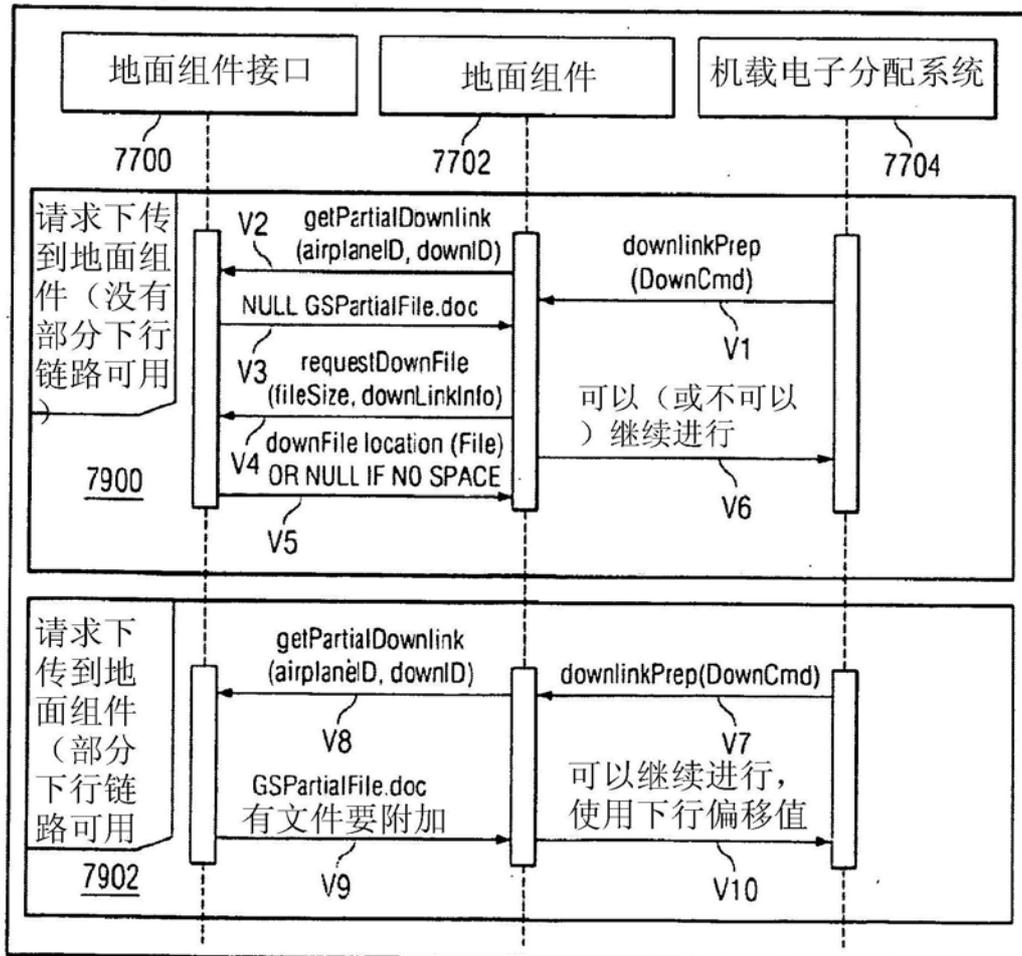


图79

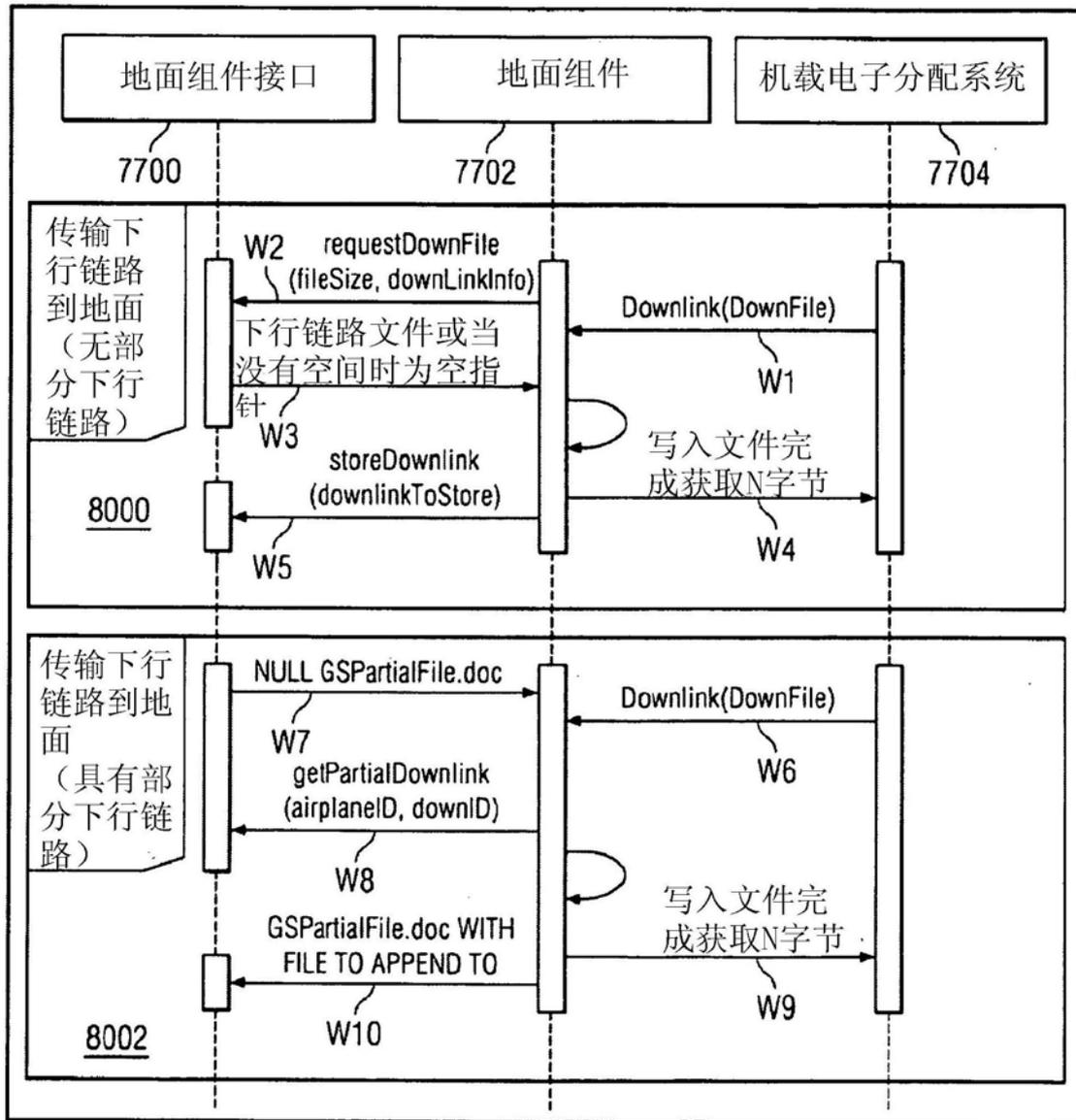


图80

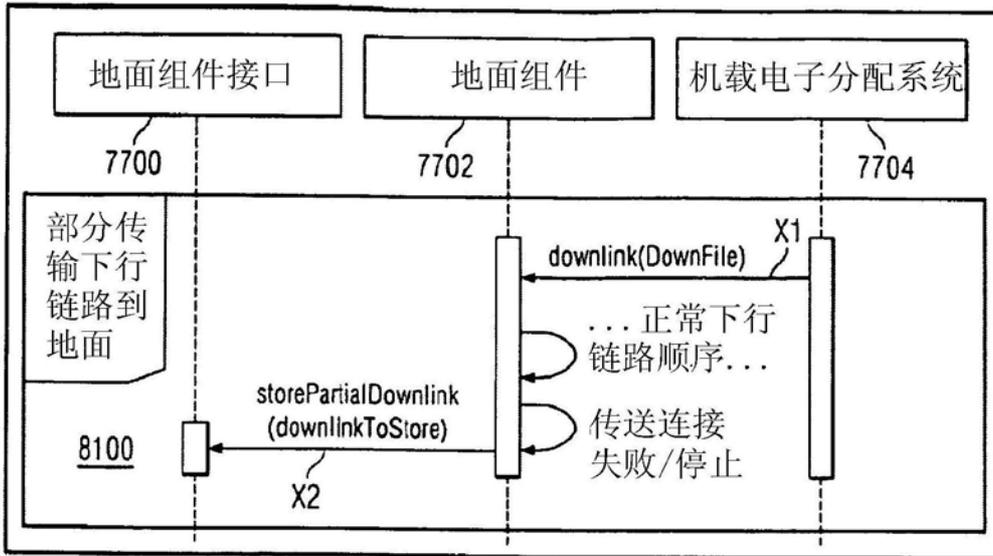


图81

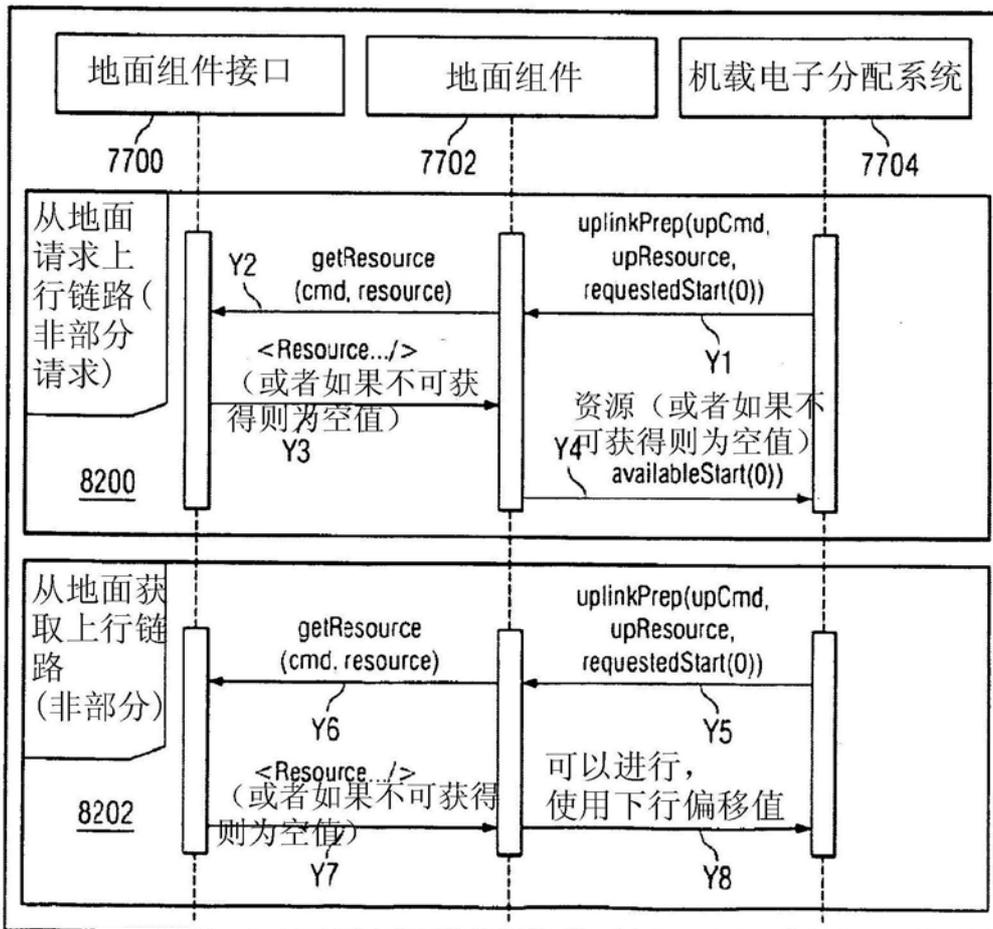


图82

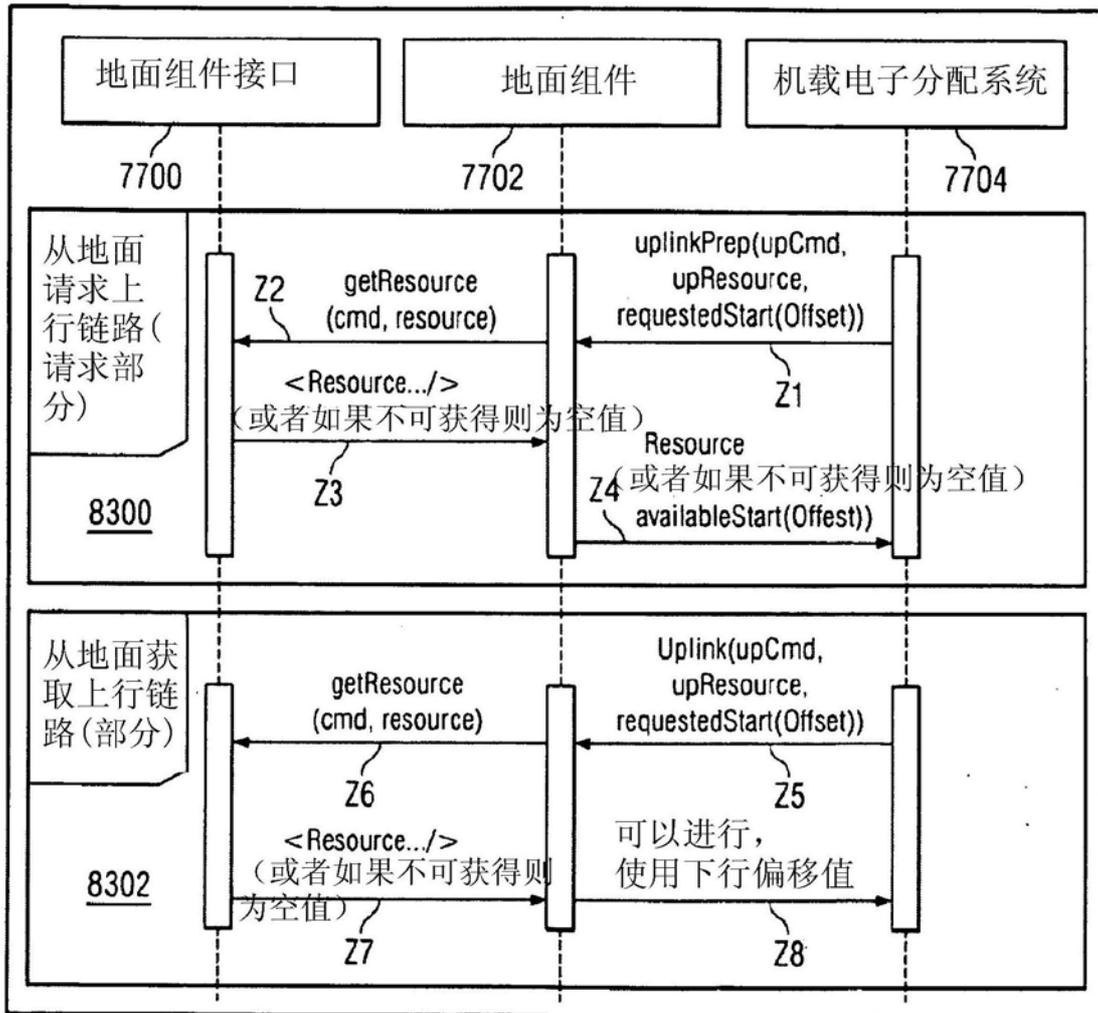


图83

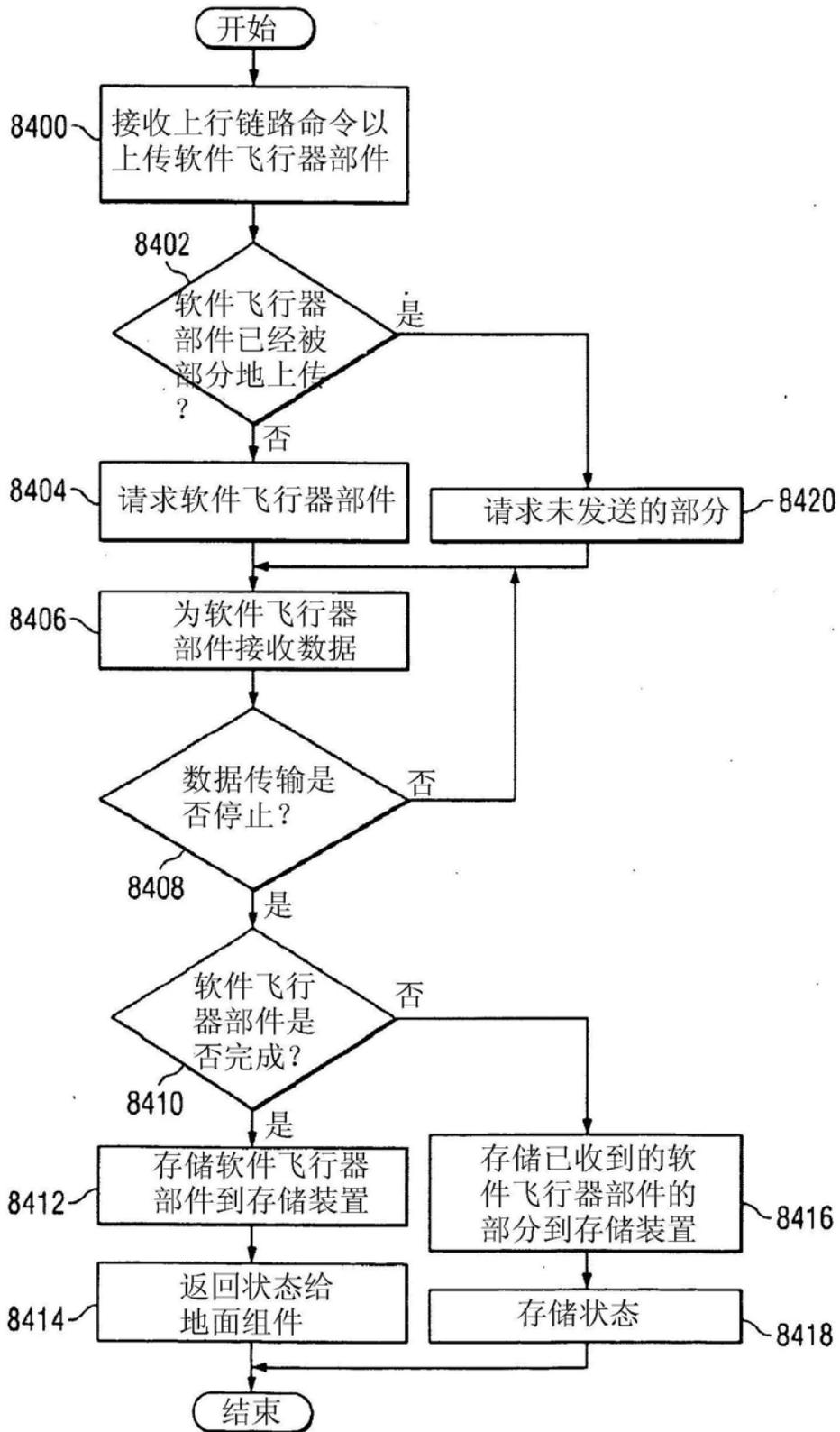


图84

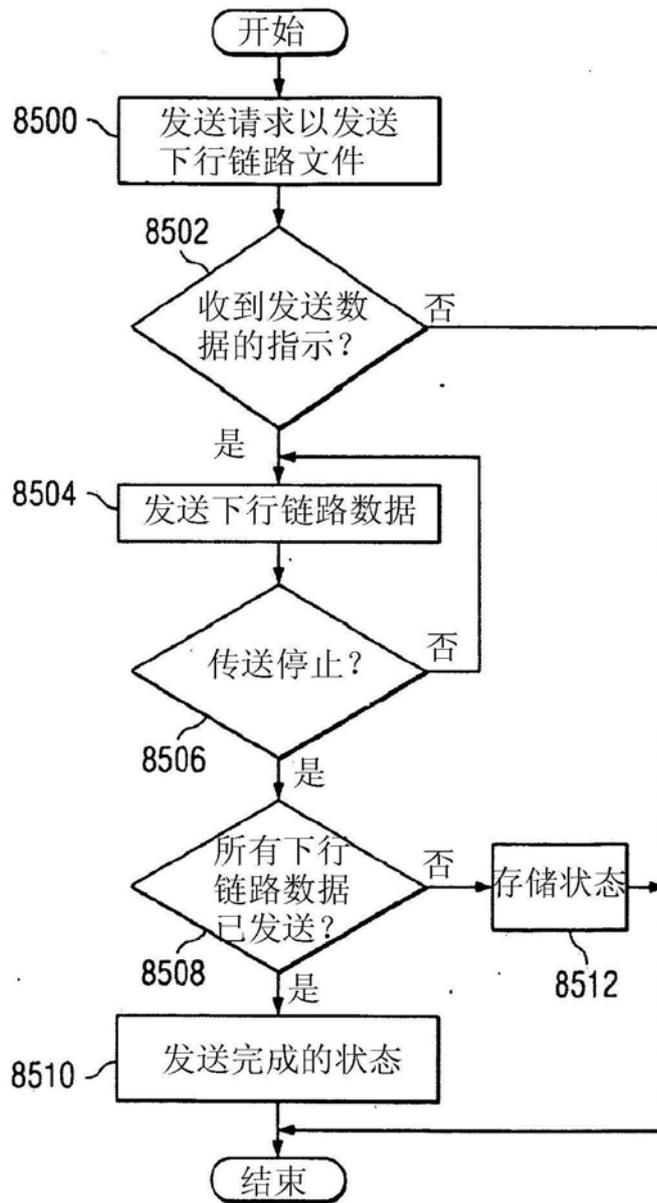


图85

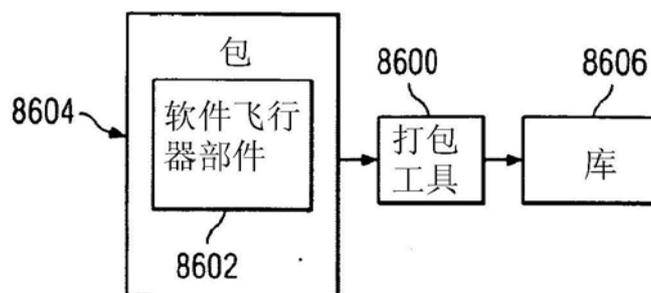


图86

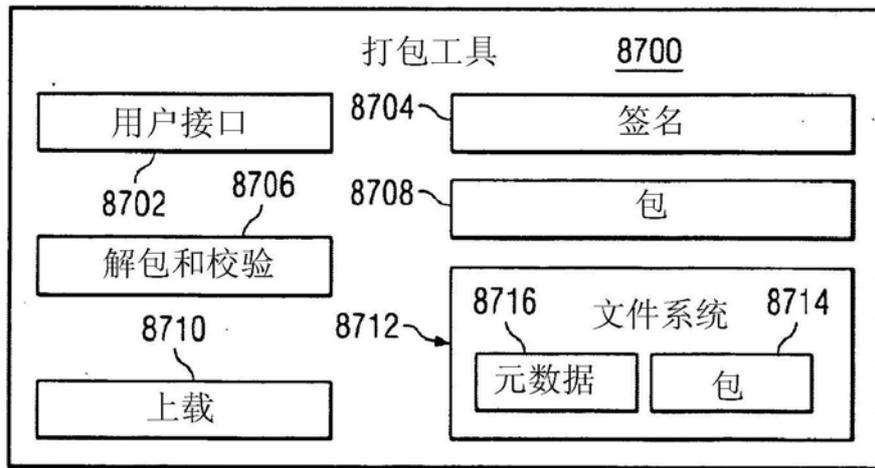


图87

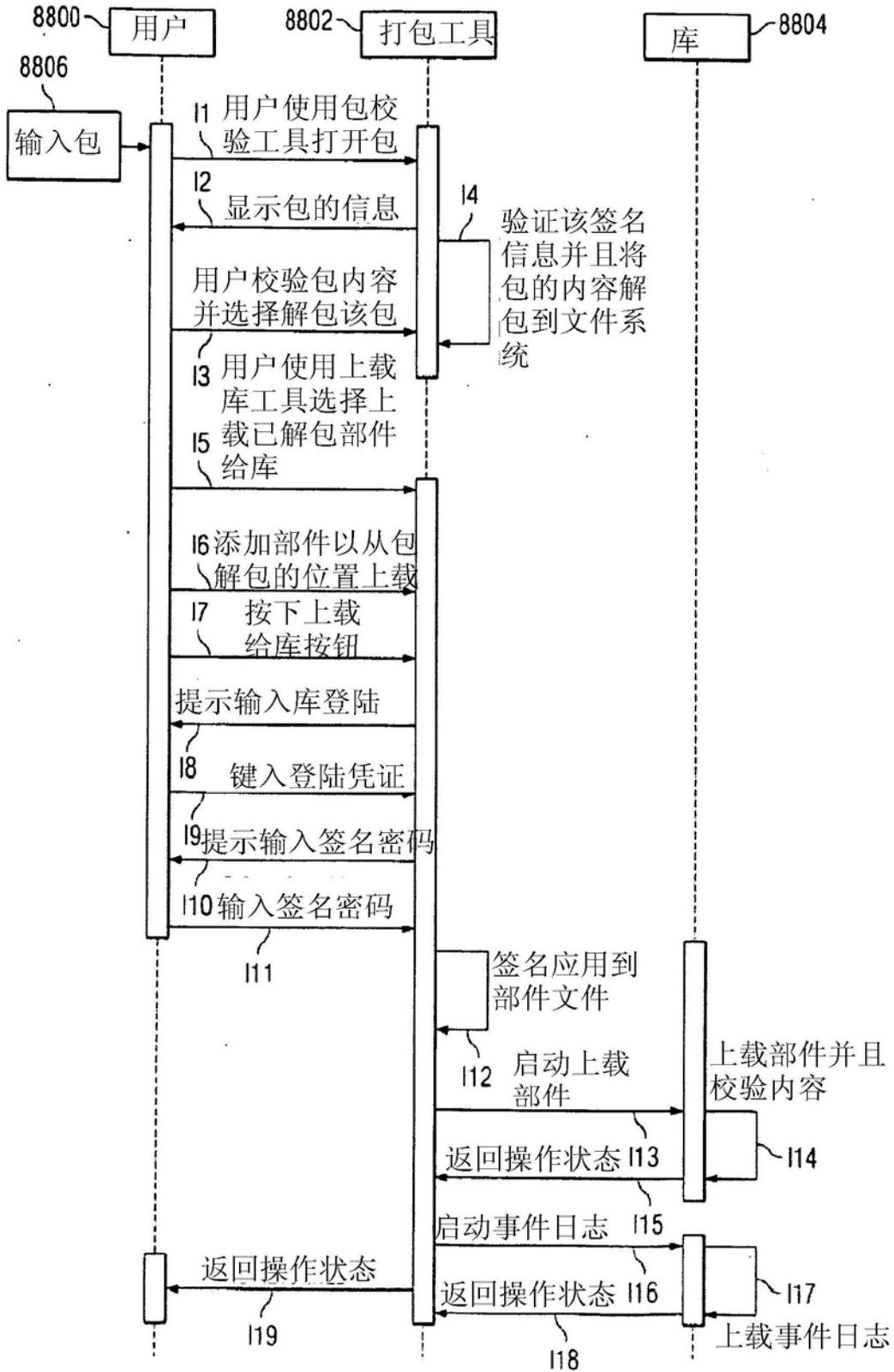


图88

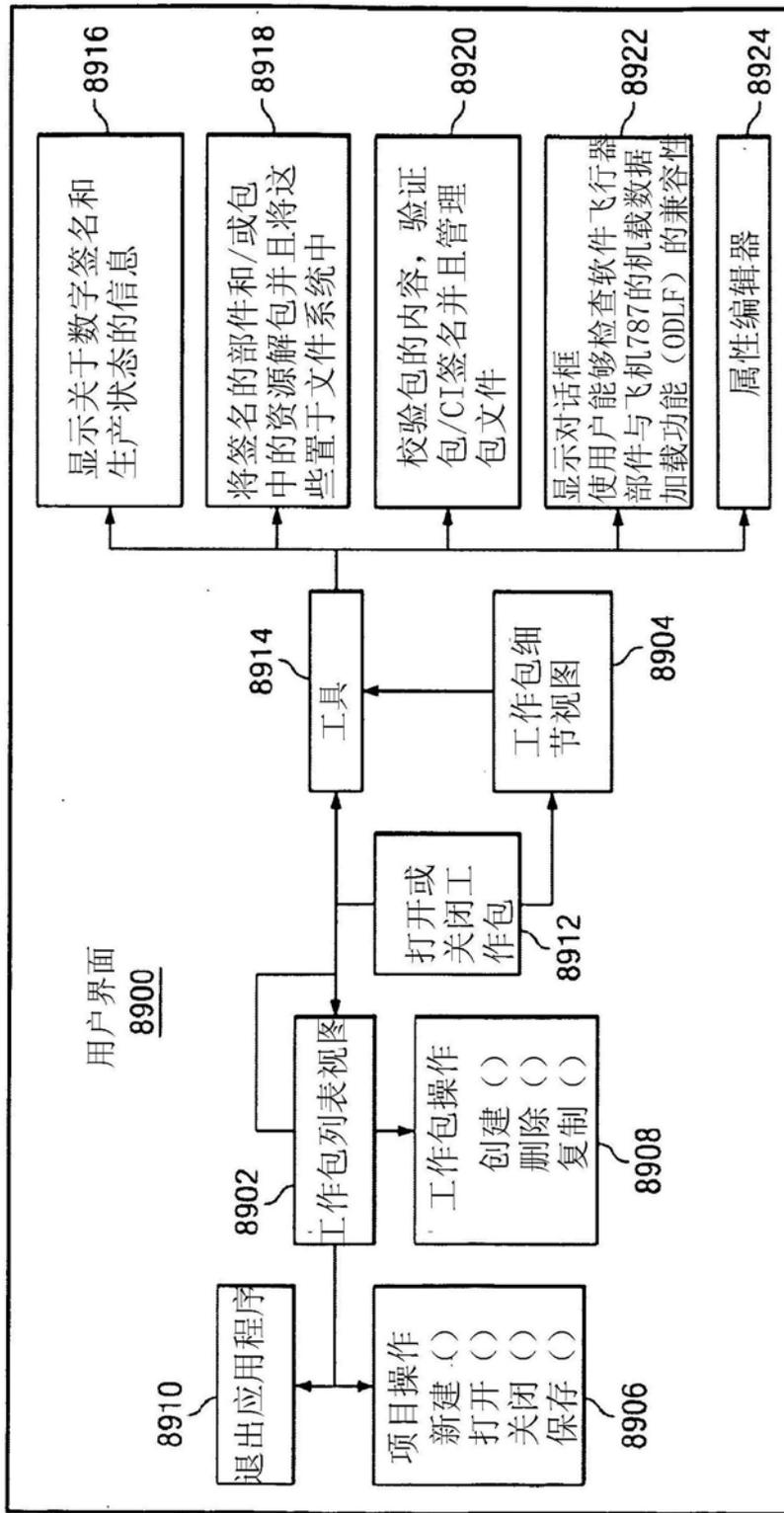


图89

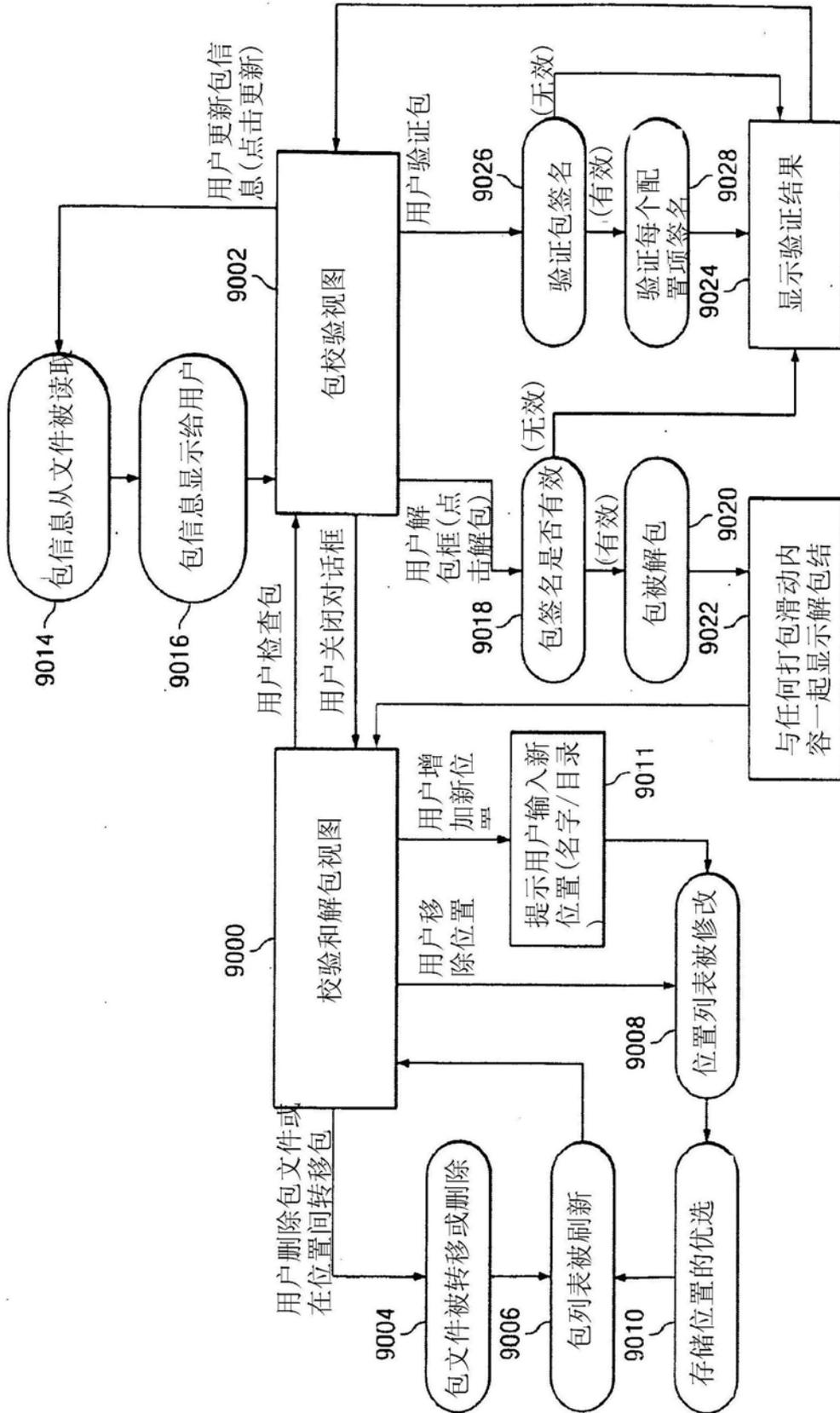


图90

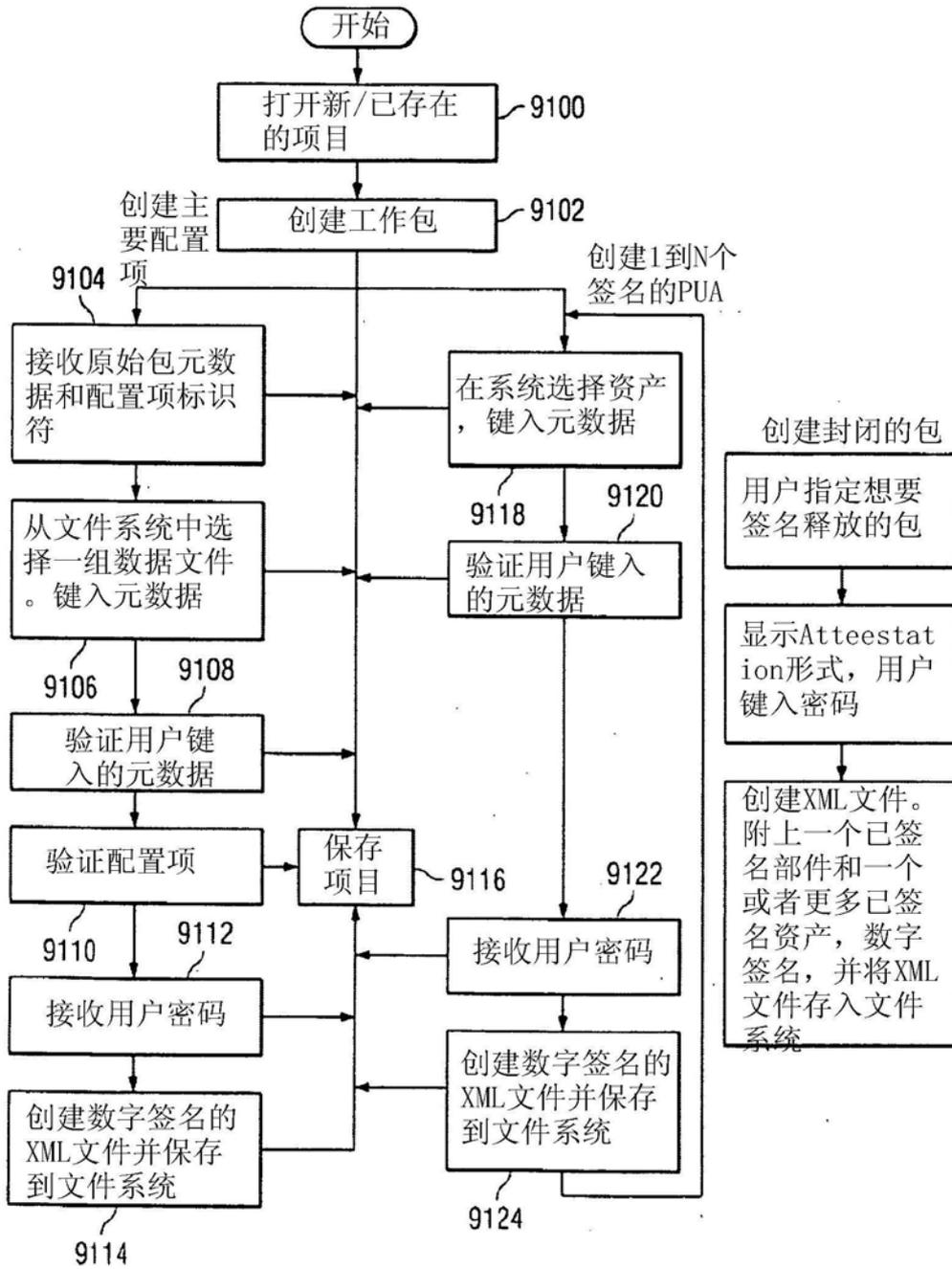


图91

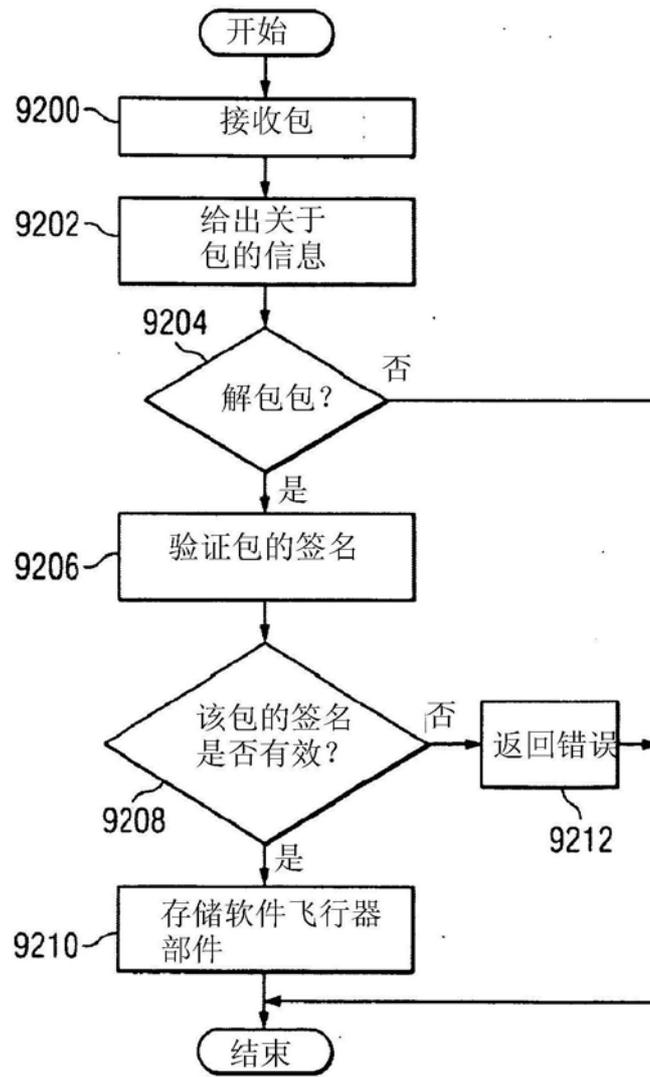


图92