



(12)发明专利申请

(10)申请公布号 CN 110895460 A

(43)申请公布日 2020.03.20

(21)申请号 201811066964.0

(22)申请日 2018.09.13

(71)申请人 深圳市优必选科技有限公司
地址 518000 广东省深圳市南山区学苑大道1001号南山智园C1栋16、22楼

(72)发明人 熊友军 蒋永俊

(74)专利代理机构 深圳中一联合知识产权代理有限公司 44414
代理人 张全文

(51) Int. Cl.
G06F 8/10(2018.01)
G06F 8/20(2018.01)
G06F 8/41(2018.01)

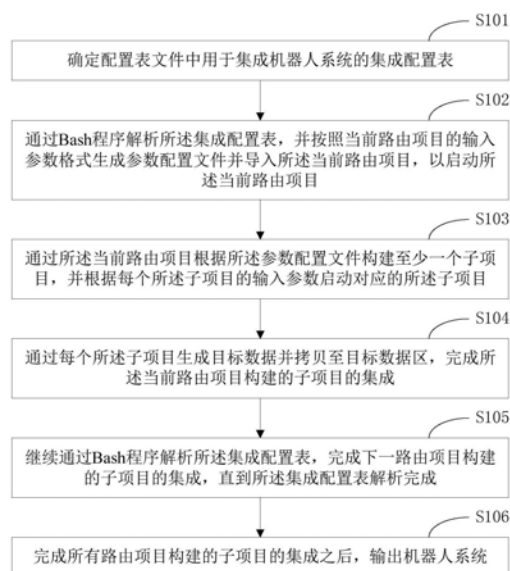
权利要求书2页 说明书8页 附图4页

(54)发明名称

基于Jenkins的机器人系统集成方法、装置及终端设备

(57)摘要

本发明适用于Jenkins技术领域,提供了一种基于Jenkins的机器人系统集成方法、系统及终端设备。本发明通过Bash程序解析用于集成机器人系统的集成配置表,并按照当前路由项目的输入参数格式,生成参数配置文件并导入当前路由项目;在当前路由项目启动后,通过当前路由项目根据参数配置文件构建至少一个子项目并启动,通过子项目生成目标数据并拷贝至目标数据区,以完成当前路由项目构建的子项目的集成;然后按照上述步骤完成下一路由项目对应的子项目的集成,直到完成所有子项目的集成之后,输出机器人系统,可以满足集成系统中多变的配置参数的构建需求,尤其适用于集成系统的子项目是变化的情况。



1. 一种基于Jenkins的机器人系统集成方法,其特征在于,包括:
 - 确定配置表文件中用于集成机器人系统的集成配置表;
 - 通过Bash程序解析所述集成配置表,并按照当前路由项目的输入参数格式生成参数配置文件并导入所述当前路由项目,以启动所述当前路由项目;
 - 通过所述当前路由项目根据所述参数配置文件构建至少一个子项目,并根据每个所述子项目的输入参数启动对应的所述子项目;
 - 通过每个所述子项目生成目标数据并拷贝至目标数据区,完成所述当前路由项目构建的子项目的集成;
 - 继续通过Bash程序解析所述集成配置表,完成下一路由项目构建的子项目的集成,直到所述集成配置表解析完成;
 - 完成所有路由项目构建的子项目的集成之后,输出机器人系统。
2. 如权利要求1所述的基于Jenkins的机器人系统集成方法,其特征在于,通过每个所述子项目生成目标数据并拷贝至目标数据区,完成所述当前路由项目构建的子项目的集成,包括:
 - 通过每个所述子项目根据所述参数配置文件生成目标数据并拷贝至目标数据区,完成所述当前路由项目构建的子项目的集成。
3. 如权利要求1所述的基于Jenkins的机器人系统集成方法,其特征在于,通过每个所述子项目生成目标数据并拷贝至目标数据区,完成所述当前路由项目构建的子项目的集成,包括:
 - 通过每个所述子项目根据所述参数配置文件和上一路由项目构建的子项目生成的目标数据,生成目标数据并拷贝至目标数据区,完成所述当前路由项目构建的子项目的集成。
4. 如权利要求1~3任一项所述的基于Jenkins的机器人系统集成方法,其特征在于,通过Bash程序解析所述集成配置表,并按照当前路由项目的输入参数格式生成参数配置文件并导入所述当前路由项目,以启动所述当前路由项目,包括:
 - 通过Bash程序解析所述集成配置表中与当前路由项目对应的数据;
 - 通过Bash程序按照所述当前路由项目的输入参数格式,将与所述当前路由项目对应的数据生成参数配置文件并导入所述当前路由项目,以启动所述当前路由项目。
5. 如权利要求1~3任一项所述的基于Jenkins的机器人系统集成方法,其特征在于,所述集成配置表包括用于构建所述集成机器人系统的构建产品名、仓库分支、系统版本和编译模式中的至少一种,还包括用于构建所述集成机器人系统的多组不同配置的应用组的应用名列表和/或应用仓库构件分支。
6. 一种基于Jenkins的机器人系统集成装置,其特征在于,包括:
 - 确定模块,用于确定配置表文件中用于集成机器人系统的集成配置表;
 - 解析模块,用于通过Bash程序解析所述集成配置表,并按照当前路由项目的输入参数格式生成参数配置文件并导入所述当前路由项目,以启动所述当前路由项目;
 - 构建模块,用于通过所述当前路由项目根据所述参数配置文件构建至少一个子项目,并根据每个所述子项目的输入参数启动对应的所述子项目;
 - 生成模块,用于通过每个所述子项目生成目标数据并拷贝至目标数据区,完成所述当前路由项目构建的子项目的集成;

所述解析模块还用于继续通过Bash程序解析所述集成配置表,完成下一路由项目构建的子项目的集成,直到所述集成配置表解析完成;

输出模块,用于完成所有路由项目构建的子项目的集成之后,输出机器人系统。

7.如权利要求6所述的基于Jenkins的机器人系统集成装置,其特征在于,所述生成模块包括:

第一生成单元,用于通过每个所述子项目根据所述参数配置文件生成目标数据并拷贝至目标数据区,完成所述当前路由项目构建的子项目的集成。

8.如权利要求6所述的基于Jenkins的机器人系统集成装置,其特征在于,所述生成模块包括:

第二生成单元,用于通过每个所述子项目根据所述参数配置文件和上一路由项目构建的子项目生成的目标数据,生成目标数据并拷贝至目标数据区,完成所述当前路由项目构建的子项目的集成。

9.一种终端设备,包括存储器、处理器以及存储在所述存储器中并可在所述处理器上运行的计算机程序,其特征在于,所述处理器执行所述计算机程序时实现如权利要求1至5任一项所述方法的步骤。

10.一种计算机可读存储介质,所述计算机可读存储介质存储有计算机程序,其特征在于,所述计算机程序被处理器执行时实现上述方法的步骤。

基于Jenkins的机器人系统集成方法、装置及终端设备

技术领域

[0001] 本发明属于Jenkins技术领域,尤其涉及一种基于Jenkins的机器人系统集成方法、系统及终端设备。

背景技术

[0002] Jenkins是一个开源软件项目,是基于Java开发的一种持续集成工具,用于监控持续重复的工作,旨在提供一个开放易用的软件平台,使软件的持续集成变成可能。Jenkins在自动化的构建集成软件系统方面拥有一些基础的功能,可实现软件系统构建过程中的项目启动控制、Bash程序执行、项目触发等功能。

[0003] 然而,使用Jenkins项目来构建配置参数的操作较为死板,不能满足系统集成过程中多变的配置参数的构建需求,尤其无法适用于系统集成的子项目是变化的情况,使用Jenkins配置框架来集成复杂系统,操作复杂甚至不能实现。

发明内容

[0004] 有鉴于此,本发明实施例提供了一种基于Jenkins的机器人系统集成方法、系统及终端设备,以解决现有技术中使用Jenkins项目来构建配置参数的操作较为死板,不能满足集成系统中多变的配置参数的构建需求,尤其无法适用于集成系统的子项目是变化的情况,使用Jenkins配置框架来集成复杂系统,操作复杂甚至不能实现的问题。

[0005] 本发明实施例的第一方面提供了一种基于Jenkins的机器人系统集成方法,其包括:

[0006] 确定配置表文件中用于集成机器人系统的集成配置表;

[0007] 通过Bash程序解析所述集成配置表,并按照当前路由项目的输入参数格式生成参数配置文件并导入所述当前路由项目,以启动所述当前路由项目;

[0008] 通过所述当前路由项目根据所述参数配置文件构建至少一个子项目,并根据每个所述子项目的输入参数启动对应的所述子项目;

[0009] 通过每个所述子项目生成目标数据并拷贝至目标数据区,完成所述当前路由项目构建的子项目的集成;

[0010] 继续通过Bash程序解析所述集成配置表,完成下一路由项目构建的子项目的集成,直到所述集成配置表解析完成;

[0011] 完成所有路由项目构建的子项目的集成之后,输出机器人系统。

[0012] 本发明实施例的第二方面提供了一种基于Jenkins的机器人系统集成装置,其包括:

[0013] 确定模块,用于确定配置表文件中用于集成机器人系统的集成配置表;

[0014] 解析模块,用于通过Bash程序解析所述集成配置表,并按照当前路由项目的输入参数格式生成参数配置文件并导入所述当前路由项目,以启动所述当前路由项目;

[0015] 构建模块,用于通过所述当前路由项目根据所述参数配置文件构建至少一个子项

目,并根据每个所述子项目的输入参数启动对应的所述子项目;

[0016] 生成模块,用于通过每个所述子项目生成目标数据并拷贝至目标数据区,完成所述当前路由项目构建的子项目的集成;

[0017] 所述解析模块还用于继续通过Bash程序解析所述集成配置表,完成下一路由项目构建的子项目的集成,直到所述集成配置表解析完成;

[0018] 输出模块,用于完成所有路由项目构建的子项目的集成之后,输出机器人系统。

[0019] 本发明实施例的第三方面提供了一种终端设备,包括存储器、处理器以及存储在所述存储器中并可在所述处理器上运行的计算机程序,所述处理器执行所述计算机程序时实现上述方法的步骤。

[0020] 本发明实施例的第四方面提供了一种计算机可读存储介质,所述计算机可读存储介质存储有计算机程序,所述计算机程序被处理器执行时实现上述方法的步骤。

[0021] 本发明实施例在Jenkins的当前路由项目构建子项目之前,通过Bash程序解析用于集成机器人系统的集成配置表,并按照当前路由项目的输入参数格式,生成参数配置文件并导入当前路由项目,以启动当前路由项目;在当前路由项目启动后,通过当前路由项目根据参数配置文件构建至少一个子项目并启动,通过子项目生成目标数据并拷贝至目标数据区,以完成当前路由项目构建的子项目的集成;然后再通过Bash程序继续解析集成配置表并按照上述步骤完成下一路由项目对应的子项目的集成,直到完成所有子项目的集成之后,输出机器人系统,可以满足集成系统中多变的配置参数的构建需求,尤其适用于集成系统的子项目是变化的情况,可以用来集成复杂系统,操作简单、易于实现且集成效率高,利于维护、成本低。

附图说明

[0022] 为了更清楚地说明本发明实施例中的技术方案,下面将对实施例或现有技术描述中所需要使用的附图作简单地介绍,显而易见地,下面描述中的附图仅仅是本发明的一些实施例,对于本领域普通技术人员来讲,在不付出创造性劳动性的前提下,还可以根据这些附图获得其他的附图。

[0023] 图1是本发明实施例一提供的基于Jenkins的机器人系统集成方法的流程示意图;

[0024] 图2是本发明实施例一提供的集成配置表的示意图;

[0025] 图3是本发明实施例一提供的基于Jenkins的机器人系统集成架构的示意图;

[0026] 图4是本发明实施例二提供的基于Jenkins的机器人系统集成装置的结构示意图;

[0027] 图5是本发明实施例三提供的终端设备的结构示意图。

具体实施方式

[0028] 为了使本技术领域的人员更好地理解本发明方案,下面将结合本发明实施例中的附图,对本发明实施例中的技术方案进行清楚地描述,显然,所描述的实施例是本发明一部分的实施例,而不是全部的实施例。基于本发明中的实施例,本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例,都应当属于本发明保护的范围。

[0029] 本发明的说明书和权利要求书及上述附图中的术语“包括”以及它们任何变形,意图在于覆盖不排他的包含。例如包含一系列步骤或单元的过程、方法或系统、产品或设备没

有限定于已列出的步骤或单元,而是可选地还包括没有列出的步骤或单元,或可选地还包括对于这些过程、方法、产品或设备固有的其它步骤或单元。此外,术语“第一”、“第二”和“第三”等是用于区别不同对象,而非用于描述特定顺序。

[0030] 实施例一

[0031] 本实施例提供一种基于Jenkins的机器人系统集成方法,其可以应用于任意的能够集成基于Jenkins的软件系统的终端设备,例如,桌上型计算机、笔记本、掌上电脑系统集成专用设备、服务器等计算设备。

[0032] 在本实施例中,基于Jenkins的机器人系统具体是指应用于机器人的安卓(Android)系统。

[0033] 如图1所示,本实施例所提供的基于Jenkins的机器人系统集成方法,包括:

[0034] 步骤S101、确定配置表文件中用于集成机器人系统的集成配置表。

[0035] 在具体应用中,配置表文件是在Jenkins配置的基础框架下构建的配置表文件,在构建配置表文件之后,需要通过PackageManager确定该配置表文件中用于集成机器人系统的集成配置表,以进入机器人系统集成步骤。

[0036] 在一个实施例中,步骤S101具体包括:

[0037] 通过PackageManager确定配置表文件中用于集成机器人系统的集成配置表。

[0038] 在本实施例中,PackageManager是用于集成机器人系统的总控制项目,它的主要职责是管理集成机器人系统的程序包,PackageManager可以从位于集成机器人系统的程序包的根目录的全局配置文件中获取并确定用于集成机器人系统的集成配置表。

[0039] 在一个实施例中,所述集成配置表包括用于构建所述集成机器人系统的构建产品名、仓库分支、系统版本和编译模式中的至少一种,还包括用于构建所述集成机器人系统的多组不同配置的应用组的应用名列表和/或应用仓库构件分支。

[0040] 在具体应用中,应用组的每个应用对应一个子项目,每组应用组都对应有用于集成该组应用的应用名列表、应用仓库构件分支等信息。

[0041] 步骤S102、通过Bash程序解析所述集成配置表,并按照当前路由项目的输入参数格式生成参数配置文件并导入所述当前路由项目,以启动所述当前路由项目。

[0042] 在具体应用中,Bash程序即为Shell程序,其本身是一种脚本语言,可以根据实际需要解释和执行相应的脚本文件,以执行系统调用或系统命令。本实施例中,Bash程序用于解析集成配置表。

[0043] 在具体应用中,路由项目是指利用Jenkins构建的路由项目,一个集成系统对应至少一个路由项目。本实施例中,机器人系统对应一个以上路由项目,当前路由项目是指Bash程序当前执行解析操作的对象。

[0044] 在一个实施例中,步骤S102具体包括:

[0045] 通过Bash程序解析所述集成配置表中与当前路由项目对应的数据;

[0046] 通过Bash程序按照所述当前路由项目的输入参数格式,将与所述当前路由项目对应的数据生成参数配置文件并导入所述当前路由项目,以启动所述当前路由项目。

[0047] 在具体应用中,集成配置表包括机器人系统的配置参数以及与每个路由项目对应的用于构建子项目的配置数据。在构建当前路由项目对应的子项目之前,通过Bash程序解析集成配置表中用于构建当前路由项目对应的子项目的数据,然后按照当前路由项目的输

入参数格式,将该数据生成一个参数配置文件并导入当前路由项目,以启动当前路由项目,使当前路由项目可以根据参数配置文件构建子项目。参数配置文件还可以包括集成配置表中机器人系统的配置参数。

[0048] 如图2所示,示例性的示出了一个用于集成机器人系统的集成配置表;其中,head区配置表示机器人系统的配置参数,配置参数包含了构建机器人系统所需的构建产品名、仓库分支、系统版本、编译模式等信息;group区有多组,对应需要构建集成的多组不同配置的应用组,每组group都包含集成其对应的应用组所需的应用名列表、应用仓库构件分支等信息。这些信息都会通过Bash程序从集成配置表中解析出来生成参数配置文件。每个应用组包括至少一个应用,每个应用对应一个子项目。

[0049] 步骤S103、通过所述当前路由项目根据所述参数配置文件构建至少一个子项目,并根据每个所述子项目的输入参数启动对应的所述子项目。

[0050] 在具体应用中,路由项目和子项目是Jenkins中用于集成系统的工具,这些工具本身就是客观存在的,在Bash程序解析配置表生成与一个路由项目对应的参数配置文件之后,路由项目即根据参数配置文件调用相应的子项目并启动,以在子项目运行之后输出构成集成系统的目标数据。路由项目构建的子项目的数据是由参数配置文件来决定的,可以只构建一个子项目也可以构建多个子项目,多个子项目的构建顺序也是由参数配置文件来决定的,多个子项目可以被依次构建,也可以同时构建。

[0051] 步骤S104、通过每个所述子项目生成目标数据并拷贝至目标数据区,完成所述当前路由项目构建的子项目的集成。

[0052] 在具体应用中,子项目启动运行之后即输出用于集成系统的目标数据,该目标数据即为集成系统的一部分。目标数据区是用于存储目标数据的数据仓库。目标数据的生成顺序由子项目的构建顺序和子项目的运行时间决定,即先运行完成的子项目,先生成目标数据。通常情况下,先构建并启动的子项目,先运行完成。

[0053] 在具体应用中,目标数据包括子项目启动运行后生成的应用程序代码和/或测试报告数据。用户可以通过访问目标数据区来查看测试报告数据,从而判断系统是否集成成功。

[0054] 在一个实施例中,步骤S104具体包括:

[0055] 通过每个所述子项目根据所述参数配置文件生成目标数据并拷贝至目标数据区,完成所述当前路由项目构建的子项目的集成。

[0056] 在具体应用中,子项目启动运行时可以根据参数配置文件中的信息生成目标数据,子项目自身的配置文件中配置有参数配置文件的访问路径,使子项目可以访问参数配置文件。

[0057] 步骤S105、继续通过Bash程序解析所述集成配置表,完成下一路由项目构建的子项目的集成,直到所述集成配置表解析完成。

[0058] 在具体应用中,在完成了一个路由项目构建的所有子项目的集成之后,重复执行步骤S102-S104,完成下一路由项目构建的子项目的集成,直到所述集成配置表解析完成,则不再重复执行。即每个路由项目对应的子项目的集成步骤均与步骤S102-S104相同。

[0059] 在一个实施例中,步骤S104具体包括:

[0060] 通过每个所述子项目根据所述参数配置文件和上一路由项目构建的子项目生成

的目标数据,生成目标数据并拷贝至目标数据区,完成所述当前路由项目构建的子项目的集成。

[0061] 在具体应用中,当前路由项目为非首个路由项目时,其构建的子项目,可以根据当前路由项目对应的参数配置文件和上一路由项目构建的子项目生成的目标数据,来生成目标数据。

[0062] 在一个实施例中,步骤S102之前包括:

[0063] 启动Bash程序;

[0064] 对应的,步骤S105之后包括:

[0065] 退出所述Bash程序。

[0066] 步骤S106、所有路由项目构建的子项目的集成之后,输出机器人系统。

[0067] 在具体应用中,完成所有路由项目的子项目的集成之后,即完成了机器人系统的集成操作,得到完整的应用于机器人的机器人系统。

[0068] 如图3所示,示例性的示出了路由项目的数量为两个且每个路由项目均构建多个子项目时,实现基于Jenkins的机器人系统集成方法的流程示意图。其中,路由项目1对应的参数配置文件表示为参数配置文件A,路由项目1构建的多个子项目表示为子项目A、子项目B、……、子项目N,路由项目1构建的多个子项目生成的目标数据拷贝至目标数据区A;路由项目2对应的参数配置文件表示为参数配置文件B,路由项目2构建的多个子项目表示为子项目a、子项目b、……、子项目n,路由项目2构建的多个子项目生成的目标数据拷贝至目标数据区B。

[0069] 图3中虚线表示可能执行的流程,即路由项目1构建的多个子项目在启动后可以根据参数配置文件A中的配置数据来生成目标数据;路由项目2构建的多个子项目可以在启动后可以根据参数配置文件A中的配置数据和目标数据区A中的目标数据来生成目标数据。

[0070] 应理解,上述实施例中各步骤的序号的大小并不意味着执行顺序的先后,各过程的执行顺序应以其功能和内在逻辑确定,而不对本发明实施例的实施过程构成任何限定。

[0071] 本实施例在Jenkins的当前路由项目构建子项目之前,通过Bash程序解析用于集成机器人系统的集成配置表,并按照当前路由项目的输入参数格式,生成参数配置文件并导入当前路由项目,以启动当前路由项目;在当前路由项目启动后,通过当前路由项目根据参数配置文件构建至少一个子项目并启动,通过子项目生成目标数据并拷贝至目标数据区,以完成当前路由项目构建的子项目的集成;然后再通过Bash程序继续解析集成配置表并按照上述步骤完成下一路由项目对应的子项目的集成,直到完成所有子项目的集成之后,输出机器人系统,可以满足集成系统中多变的配置参数的构建需求,尤其适用于集成系统的子项目是变化的情况,可以用来集成复杂系统,操作简单、易于实现且集成效率高,利于维护、成本低。

[0072] 实施例二

[0073] 本实施例提供一种基于Jenkins的机器人系统集成装置,用于执行实施例一中的基于Jenkins的机器人系统集成方法。该机器人系统集成装置可以是任意的能够集成基于Jenkins的软件系统的终端设备,例如,桌上型计算机、笔记本、掌上电脑系统集成专用设备、服务器等计算设备。

[0074] 如图4所示,本实施例所提供的基于Jenkins的机器人系统集成装置4,包括如下软件程序模块:

[0075] 确定模块401,用于确定配置表文件中用于集成机器人系统的集成配置表;

[0076] 解析模块402,用于通过Bash程序解析所述集成配置表,并按照当前路由项目的输入参数格式生成参数配置文件并导入所述当前路由项目,以启动所述当前路由项目;

[0077] 构建模块403,用于通过所述当前路由项目根据所述参数配置文件构建至少一个子项目,并根据每个所述子项目的输入参数启动对应的所述子项目;

[0078] 生成模块404,用于通过每个所述子项目生成目标数据并拷贝至目标数据区,完成所述当前路由项目构建的子项目的集成;

[0079] 所述解析模块402还用于继续通过Bash程序解析所述集成配置表,完成下一路由项目构建的子项目的集成,直到所述集成配置表解析完成;

[0080] 输出模块405,用于完成所有路由项目构建的子项目的集成之后,输出机器人系统。

[0081] 在一个实施例中,所述生成模块包括:

[0082] 第一生成单元,用于通过每个所述子项目根据所述参数配置文件生成目标数据并拷贝至目标数据区,完成所述当前路由项目构建的子项目的集成。

[0083] 在一个实施例中,所述生成模块包括:

[0084] 第二生成单元,用于通过每个所述子项目根据所述参数配置文件和上一路由项目构建的子项目生成的目标数据,生成目标数据并拷贝至目标数据区,完成所述当前路由项目构建的子项目的集成。

[0085] 在一个实施例中,所述基于Jenkins的机器人系统集成装置还包括:

[0086] 程序启动模块,用于启动Bash程序;

[0087] 程序退出模块,用于退出所述Bash程序。

[0088] 本实施例在Jenkins的当前路由项目构建子项目之前,通过Bash程序解析用于集成机器人系统的集成配置表,并按照当前路由项目的输入参数格式,生成参数配置文件并导入当前路由项目,以启动当前路由项目;在当前路由项目启动后,通过当前路由项目根据参数配置文件构建至少一个子项目并启动,通过子项目生成目标数据并拷贝至目标数据区,以完成当前路由项目构建的子项目的集成;然后再通过Bash程序继续解析集成配置表并按照上述步骤完成下一路由项目对应的子项目的集成,直到完成所有子项目的集成之后,输出机器人系统,可以满足集成系统中多变的配置参数的构建需求,尤其适用于集成系统的子项目是变化的情况,可以用来集成复杂系统,操作简单、易于实现且集成效率高,利于维护、成本低。

[0089] 实施例三

[0090] 如图5所示,本实施例提供一种终端设备5,其包括:处理器50、存储器51以及存储在所述存储器51中并可在所述处理器50上运行的计算机程序52,例如基于Jenkins的机器人系统集成程序。所述处理器50执行所述计算机程序52时实现上述基于Jenkins的机器人系统集成方法实施例中的步骤,例如图1所示的步骤S101至S106。或者,所述处理器50执行所述计算机程序52时实现上述各装置实施例中各模块/单元的功能,例如图4所示模块401至405的功能。

[0091] 示例性的,所述计算机程序52可以被分割成一个或多个模块/单元,所述一个或者多个模块/单元被存储在所述存储器51中,并由所述处理器50执行,以完成本发明。所述一个或多个模块/单元可以是能够完成特定功能的一系列计算机程序指令段,该指令段用于描述所述计算机程序52在所述终端设备5中的执行过程。例如,所述计算机程序52可以被分割成确定模块、解析模块、构建模块、生成模块、输出模块,各模块具体功能如下:

[0092] 确定模块,用于确定配置表文件中用于集成机器人系统的集成配置表;

[0093] 解析模块,用于通过Bash程序解析所述集成配置表,并按照当前路由项目的输入参数格式生成参数配置文件并导入所述当前路由项目,以启动所述当前路由项目;

[0094] 构建模块,用于通过所述当前路由项目根据所述参数配置文件构建至少一个子项目,并根据每个所述子项目的输入参数启动对应的所述子项目;

[0095] 生成模块,用于通过每个所述子项目生成目标数据并拷贝至目标数据区,完成所述当前路由项目构建的子项目的集成;

[0096] 所述解析模块还用于继续通过Bash程序解析所述集成配置表,完成下一路由项目构建的子项目的集成,直到所述集成配置表解析完成;

[0097] 输出模块,用于完成所有路由项目构建的子项目的集成之后,输出机器人系统。

[0098] 所述终端设备5可以是桌上型计算机、笔记本、掌上电脑及云端服务器等计算设备。所述终端设备可包括,但不限于,处理器50、存储器51。本领域技术人员可以理解,图5仅仅是终端设备5的示例,并不构成对终端设备5的限定,可以包括比图示更多或更少的部件,或者组合某些部件,或者不同的部件,例如所述终端设备还可以包括输入输出设备、网络接入设备、总线等。

[0099] 所称处理器50可以是中央处理单元(Central Processing Unit,CPU),还可以是其他通用处理器、数字信号处理器(Digital Signal Processor,DSP)、专用集成电路(Application Specific Integrated Circuit,ASIC)、现成可编程门阵列(Field-Programmable Gate Array,FPGA)或者其他可编程逻辑器件、分立门或者晶体管逻辑器件、分立硬件组件等。通用处理器可以是微处理器或者该处理器也可以是任何常规的处理器等。

[0100] 所述存储器51可以是所述终端设备5的内部存储单元,例如终端设备5的硬盘或内存。所述存储器51也可以是所述终端设备5的外部存储设备,例如所述终端设备5上配备的插接式硬盘,智能存储卡(Smart Media Card,SMC),安全数字(Secure Digital,SD)卡,闪存卡(Flash Card)等。进一步地,所述存储器51还可以既包括所述终端设备5的内部存储单元也包括外部存储设备。所述存储器51用于存储所述计算机程序以及所述终端设备所需的其他程序和数据。所述存储器51还可以用于暂时地存储已经输出或者将要输出的数据。

[0101] 所属领域的技术人员可以清楚地了解到,为了描述的方便和简洁,仅以上述各功能单元、模块的划分进行举例说明,实际应用中,可以根据需要而将上述功能分配由不同的功能单元、模块完成,即将所述装置的内部结构划分成不同的功能单元或模块,以完成以上描述的全部或者部分功能。实施例中的各功能单元、模块可以集成在一个处理单元中,也可以是各个单元单独物理存在,也可以两个或两个以上单元集成在一个单元中,上述集成的单元既可以采用硬件的形式实现,也可以采用软件功能单元的形式实现。另外,各功能单元、模块的具体名称也只是为了便于相互区分,并不用于限制本申请的保护范围。上述系统

中单元、模块的具体工作过程,可以参考前述方法实施例中的对应过程,在此不再赘述。

[0102] 在上述实施例中,对各个实施例的描述都各有侧重,某个实施例中沒有详述或记载的部分,可以参见其它实施例的相关描述。

[0103] 本领域普通技术人员可以意识到,结合本文中所公开的实施例描述的各示例的单元及算法步骤,能够以电子硬件、或者计算机软件和电子硬件的结合来实现。这些功能究竟以硬件还是软件方式来执行,取决于技术方案的特定应用和设计约束条件。专业技术人员可以对每个特定的应用来使用不同方法来实现所描述的功能,但是这种实现不应认为超出本发明的范围。

[0104] 在本发明所提供的实施例中,应该理解到,所揭露的装置/终端设备和方法,可以通过其它的方式实现。例如,以上所描述的装置/终端设备实施例仅仅是示意性的,例如,所述模块或单元的划分,仅仅为一种逻辑功能划分,实际实现时可以有另外的划分方式,例如多个单元或组件可以结合或者可以集成到另一个系统,或一些特征可以忽略,或不执行。另一点,所显示或讨论的相互之间的耦合或直接耦合或通讯连接可以通过一些接口,装置或单元的间接耦合或通讯连接,可以是电性,机械或其它的形式。

[0105] 所述作为分离部件说明的单元可以是或者也可以不是物理上分开的,作为单元显示的部件可以是或者也可以不是物理单元,即可以位于一个地方,或者也可以分布到多个网络单元上。可以根据实际的需要选择其中的部分或者全部单元来实现本实施例方案的目的。

[0106] 另外,在本发明各个实施例中的各功能单元可以集成在一个处理单元中,也可以是各个单元单独物理存在,也可以两个或两个以上单元集成在一个单元中。上述集成的单元既可以采用硬件的形式实现,也可以采用软件功能单元的形式实现。

[0107] 所述集成的模块/单元如果以软件功能单元的形式实现并作为独立的产品销售或使用,可以存储在一个计算机可读取存储介质中。基于这样的理解,本发明实现上述实施例方法中的全部或部分流程,也可以通过计算机程序来指令相关的硬件来完成,所述的计算机程序可存储于一计算机可读存储介质中,该计算机程序在被处理器执行时,可实现上述各个方法实施例的步骤。其中,所述计算机程序包括计算机程序代码,所述计算机程序代码可以为源代码形式、对象代码形式、可执行文件或某些中间形式等。所述计算机可读介质可以包括:能够携带所述计算机程序代码的任何实体或装置、记录介质、U盘、移动硬盘、磁碟、光盘、计算机存储器、只读存储器(ROM, Read-Only Memory)、随机存取存储器(RAM, Random Access Memory)、电载波信号、电信信号以及软件分发介质等。需要说明的是,所述计算机可读介质包含的内容可以根据司法管辖区内立法和专利实践的要求进行适当的增减,例如在某些司法管辖区,根据立法和专利实践,计算机可读介质不包括是电载波信号和电信信号。

[0108] 以上所述实施例仅用以说明本发明的技术方案,而非对其限制;尽管参照前述实施例对本发明进行了详细的说明,本领域的普通技术人员应当理解:其依然可以对前述各实施例所记载的技术方案进行修改,或者对其中部分技术特征进行等同替换;而这些修改或者替换,并不使相应技术方案的本质脱离本发明各实施例技术方案的精神和范围,均应包含在本发明的保护范围之内。

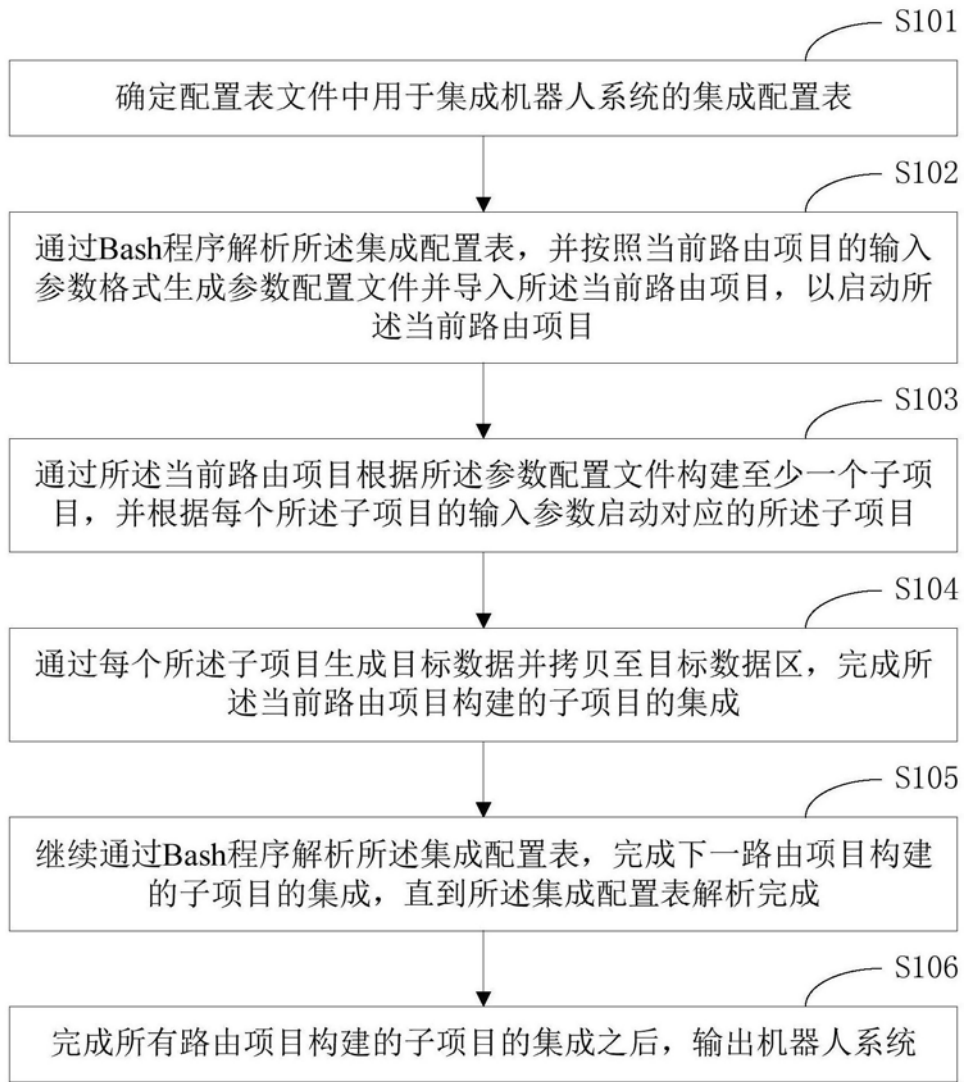


图1

```
Core-P11.1.jed
1 <head>
2     Product=Core
3     AndroidBranchOrTag=master
4     AndroidOSInfoBranch=master
5     AndroidOSVersionStart=0
6     ApkBranch=master
7     QuickCompile=true
8     VersionSign=M
9     RomCompileMode=U
10 </head>
11 <group>
12     AppsCompileGroupNameList=CruzzrMap,CruzzrTimer,CruzzrService11
13     BranchOrTag=Core_P11.1
14     IgnoreUpdate=false
15     IgnoreSDK=true
16 </group>
17 <group>
18     AppsCompileGroupNameList=CruzzrLocalQa,CruzzrWorkMode,CruzzrAiui,Cruzzr
19     SampleTest,CruzzrGenericbehavior,CruzzrApirunner,CruzzrMusic,CruzzrVide
20     o,CruzzrPowerSwitch,CruzzrRobotfacelauncher,CruzzrAlbum,CruzzrAdvert,Cr
21     uzrIntroductionGuide,CruzzrSysWarn,CruzzrDance,CruzzrPccontrolstub,Cru
22     zrUsermgr,CruzzrNavigation,CruzzrLauncher11,CruzzrProductDisplay,Cruzzr
23     BootWizard11,CruzzrFreeGuide,CruzzrOta,CruzzrPowerManager,CruzzrSetting
24     s,CruzzrSystemFaultDetection,CruzzrCanlender,CruzzrClock,CruzzrDoctor
25     BranchOrTag=Core_P11
26     IgnoreUpdate=false
27     IgnoreSDK=true
28 </group>
```

图2

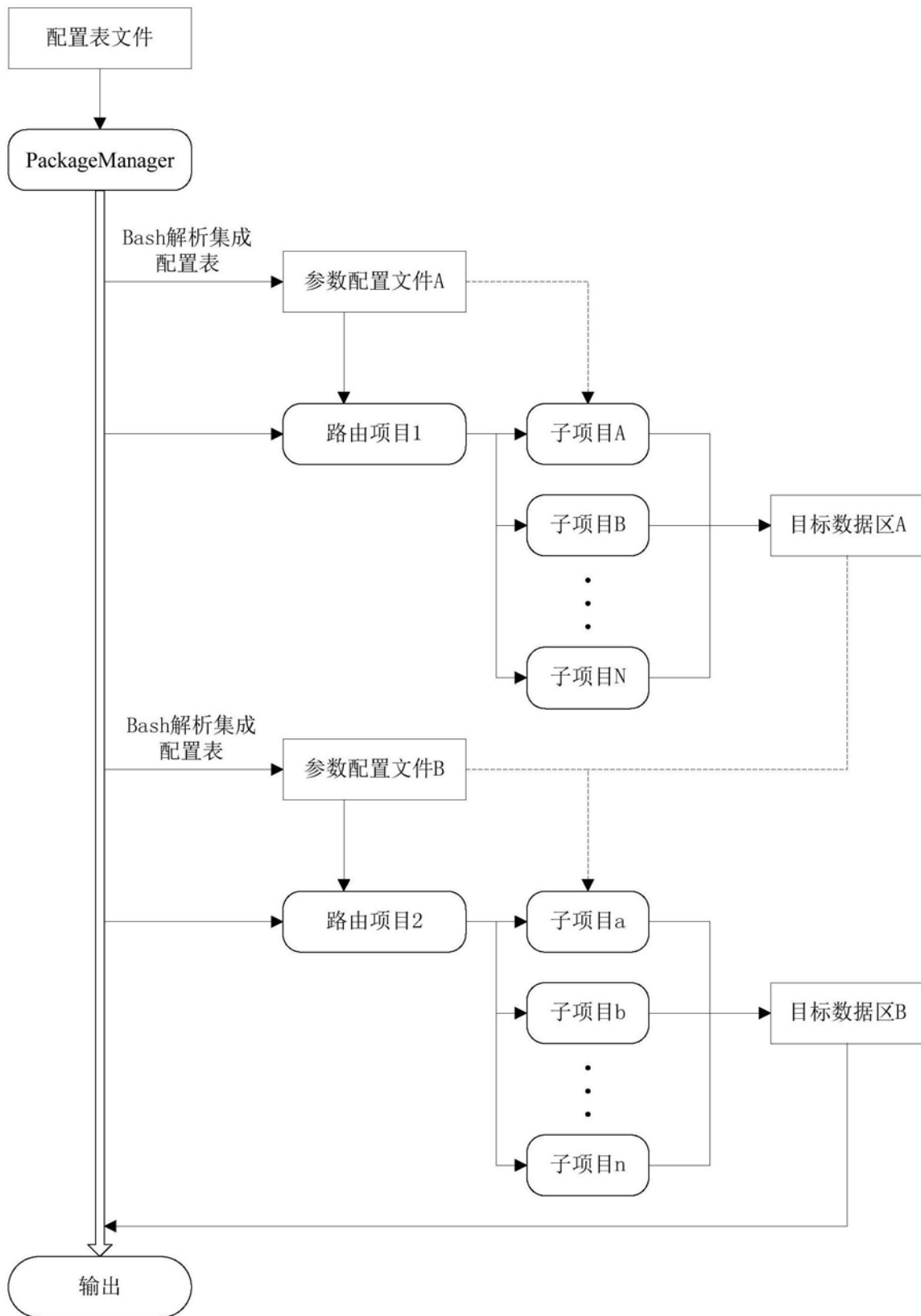


图3

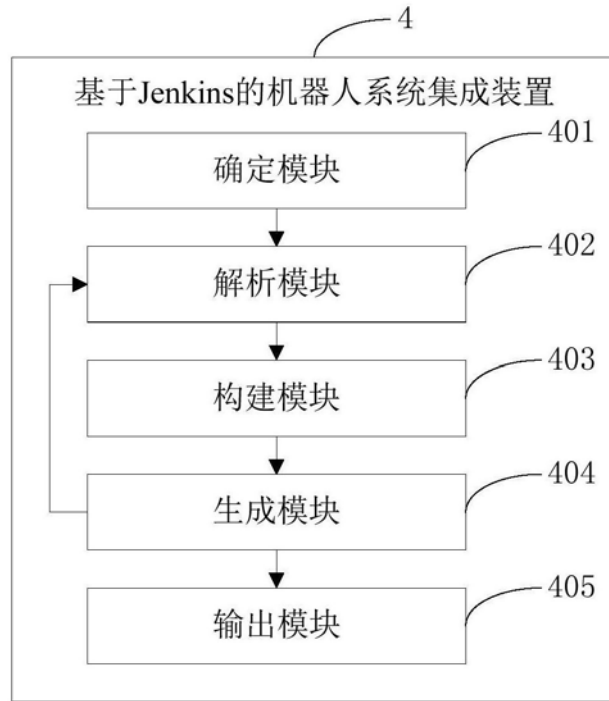


图4

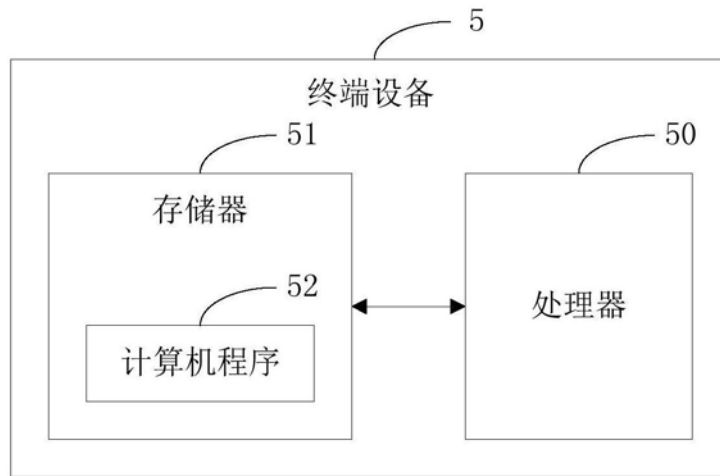


图5