

# 面向异构环境可复用的类反射式软件构件集成方法<sup>①</sup>



葛光富, 吴凯迪, 韦涛

(中国电子科技集团公司第二十八研究所, 南京 210007)

通讯作者: 葛光富, E-mail: 397559400@qq.com

**摘要:** 随着计算机领域的迅猛发展, 不断提高的软件规模及其复杂性, 导致愈加暴露的软件危机, 提高软件生产效率成了软件有关产业的当务之急. 构件化软件集成是一种解决有关问题的主要途径, 在 VxWorks 系统上可使用一种基于以函数形式定义规范软件构件操作接口的构件化软件集成方法, 该方法在工程实践中运用成熟但其存在不能满足复用于跨异构环境的需求. 故面向异构环境可复用, 本文提出了一种基于类反射机制的软件构件集成方法, 其运用工厂模式结合回调机制生成反射类基本要素, 并基于此规范软件构件类的描述与实现, 对软件构件的集成进行全生命周期的管理. 在该方法中, 整个过程以跨平台通用代码语言程序化实现使得其能够更便捷地复用于各类业务信息系统, 同时设计的软件构件基类与软件构件集成管理类结构完整、描述清晰、规范易用, 能够良好地支持构件的功能适配、更新以及软件系统的组装演化. 实验表明, 本文方法不仅适用于 VxWorks、Android、Windows 系统, 而且能够适用于 ReWorks、AOS、Kylin 等国产系统环境.

**关键词:** 软件集成; 构件; 类反射; 异构环境; 复用

引用格式: 葛光富, 吴凯迪, 韦涛. 面向异构环境可复用的类反射式软件构件集成方法. 计算机系统应用, 2021, 30(7): 197-203. <http://www.c-s-a.org.cn/1003-3254/8002.html>

## Software Integration Method of Reusable Class Reflective Component for Heterogeneous Environments

GE Guang-Fu, WU Kai-Di, WEI Tao

(The 28th Research Institute of China Electronics Technology Group Corporation, Nanjing 210007, China)

**Abstract:** With the rapid development of the computer field, the increase in software size and complexity leads to more exposed software crises. As such, improving the efficiency of software production has become an urgent task for the software-related industry. Component-based software integration is the main way to solve the related problems. On the VxWorks system, we can adopt a component-based software integration method defining standardized operation interfaces for software components in functional forms. This method is mature in engineering practice but cannot be reused in cross-heterogeneous environments. Therefore, to be reusable in the heterogeneous environments, a component-based software integration method based on class reflective mechanism is proposed, which combines the factory mode with the callback mechanism to generate the basic elements of the reflective class. On this basis, it standardizes the description and implementation of software component classes and manages the integration of software components in the whole life cycle. In this method, the whole process is implemented by the cross-platform unified code language program, which enables it to be reused more conveniently in business information systems. The designed software-component base class and integrated management class, with complete structures and clear descriptions, are standard and user-friendly and

① 基金项目: 陆装预研课题 (301010501)

Foundation item: Pre-Research Project of Land Equipment (301010501)

收稿时间: 2020-11-09; 修改时间: 2020-12-12; 采用时间: 2020-12-18; csa 在线出版时间: 2021-06-30

can well support the functional adaptation and update of components and the assembly and evolution of software systems. The experiments show that this method is suitable not only for VxWorks, Android, and Windows systems but also for domestic system environments such as ReWorks, AOS, and Kylin.

**Key words:** software integration; component; class reflective; heterogeneous environments; reusable

随着计算机领域的迅猛发展, 不断提高的软件规模及其复杂性, 导致愈加暴露的软件危机<sup>[1]</sup>, 提高软件生产效率成了软件有关产业的当务之急. 这使得软件的复用性得到了越来越多的关注, 因为通过软件复用, 可以很明显地提高软件开发的效率和质量, 软件系统的开发不必完全地从零开始, 而是可以利用一些已有的部分进行拼装<sup>[2]</sup>. 构件化软件集成是一种可高效复用软件、解决软件危机、提高软件生产效率、增加软件的开放性和对外部扰动适应性的主要途径, 它将软件项目从一个以代码编写和错误修正为中心的过程变为一个更为受控的集成过程, 软件的升级变为了替换具有良好边界的软件功能单元的过程.

利用软件构件化进行软件集成, 人们在 21 世纪初以来就开展了研究探索和实践, 提出了有关的框架设计与思路方法<sup>[3-16]</sup>, 但这些方法是针对 VxWorks、Windows 等某特定单一系统平台进行的浅适应性构件化软件集成研究, 而未对面向异构多平台环境可复用的构件化软件集成通用方法进行深入性的研究. 而在军民用市场上, 尤其是陆军领域, 有着大量的计算处理设备用于搭建各型电子与信息系统. 这些设备却部署着种类多样、复杂不一的操作系统运行环境, 其中嵌入式装备尤为突出, 如车载终端的 VxWorks、ReWorks 等, 便携终端的 Android、WinCE、AOS、SyberOS 等. 这些都无形中增加了项目软件研制开发集成的复杂性. 因此, 为支撑业务系统快速改造组装、降低项目迁移改造风险、提升应用软件系统质量, 构建跨平台运用于这些异构环境的软件构件集成技术, 成为电子与信息系统软件服务平台统筹规划中的必要考虑因素.

面向异构环境进行软件构件集成方法的通用化研究, 工程实践中如在 VxWorks 系统上有较为成熟运用的基于以函数形式规范定义构件操作接口来进而完成软件的构件化集成组装方法, 该方法按照集成规范约定构件可被集成调度用的构件初始化、启用、操作响应等接口的函数名称、参数等形式, 之后在软件运行时通过查找系统符号表来获取构件接口函数用以执行完成对应构件的初始化、启用、操作响应等, 实现软

件的构件化集成与功能调度. 故该方法仅适用于能够提供名称查找执行函数的操作系统, 而不能适用于其它如 Linux/类 Linux 等系统. 进而, 研究一种跨异构环境复用的且能够适应支撑软件集成构件清单按需配置的构件化软件集成方法, 是在跨操作系统平台应用实践中急需解决的问题.

为了解决上述难题, 本文提出了一种满足多跨平台软件研发实践需要的类反射式软件构件集成方法, 它能够对异构软硬件环境的电子与信息系统包括嵌入式装备提供可跨平台复用的软件构件集成通用手段, 支撑电子与信息系统软件的快速开发与迭代集成, 有利于降低系统开发维护成本、提升行业企业效益.

## 1 相关介绍

反射的概念是由 Smith 在 1982 年首次提出的, 主要是指一类能够自描述和自控制的应用, 这类应用通过采用某种机制来实现对自己行为的描述和监测, 并能根据自身行为的状态和结果, 调整或修改应用所描述行为的状态和相关的语义, 本文方法通过运用工厂模式结合回调机制来实现反射这一机制.

工厂模式常用于设计模式中, 没有正式统一的表述定义, 其形象比喻就是当客户需要工厂产品时通过统一向工厂下单让工厂生产后提货相应产品即可, 而不用去管产品是怎么做出来的, 如图 1(a) 所示. 本文方法中, 产品是反射类对象, 工厂是指反射类工厂.

回调机制常用于程序设计实现中, 没有正式统一的表述定义, 这一机制运用过程的形象比喻就是意为 A 让 B 做某件事时, B 做事的过程中又回过头来运用了 A 提供的方法, 如图 1(b) 所示.

软件构件为软件系统中具有相对独立功能、可以明确辨识、接口由规约指定的可组装软件实体<sup>[17]</sup>. 多个构件可被集成装载成为更高层级的软件, 在本文方法中软件构件以类的形式作为被操作对象, 具体实现中软件构件即意指软件构件类.

集成就是一些孤立的事物或元素通过某种方式改变原有的分散状态集中到一起, 产生联系, 从而构成一

个有机整体的过程<sup>[18]</sup>, 本文方法中的集成是指将所需的软件构件集合装载成为软件。

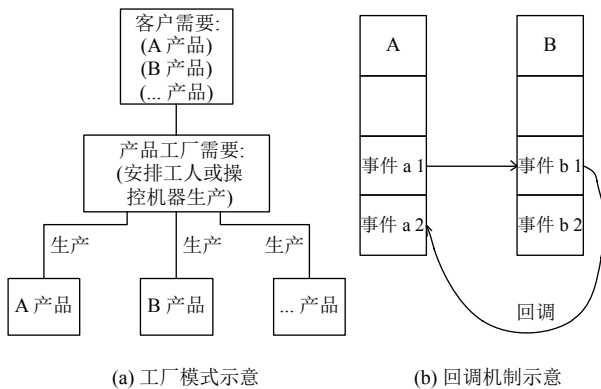


图1 工厂模式和回调机制示意

## 2 技术方案

### 2.1 整体设计

本文所述面向异构环境可复用的类反射式软件构件集成方法的整体设计包括如下过程步骤:

(1) 反射类基本要素构建. 运用工厂模式结合回调机制, 构建反射类基本要素包括反射类工厂、反射类基类、反射类通用部件。

(2) 软件构件基类构建. 运用构件化的思想, 由反射类基类派生构建软件构件基类, 支持软件构件的启停用、功能响应、心跳查询、以及其它可扩展功能。

(3) 软件构件集成管理类构建. 运用软件构件基类, 构建软件构件集成管理类, 用于集成管理软件构件的装卸载、启停用、心跳检测、故障修复。

(4) 软件构件实现. 软件构件以类的形式构建, 继承软件构件基类派生, 配装反射类通用部件, 继承软件构件基类派生, 并由构件自身实现其专有的功能。

(5) 软件按需集成管理. 业务系统软件根据自身需求预配置各自的软件构件列表, 通过运用软件构件集成管理类, 操作执行系统启动时的软件构件依序创建、启用, 运行时的软件构件功能调度、检测等, 退出时的软件构件停用、关闭, 以实现软件的构件化按需构建与运行管理。

### 2.2 反射类基本要素构建

构建的反射类基本要素包括:

#### (1) 反射类工厂

反射类工厂以反射类对象作为反射类工厂中产品, 反射类类名为产品的名称, 反射类为产品描述, 来管理所有(反射类对象)产品的生产、回收, 如图2所示。

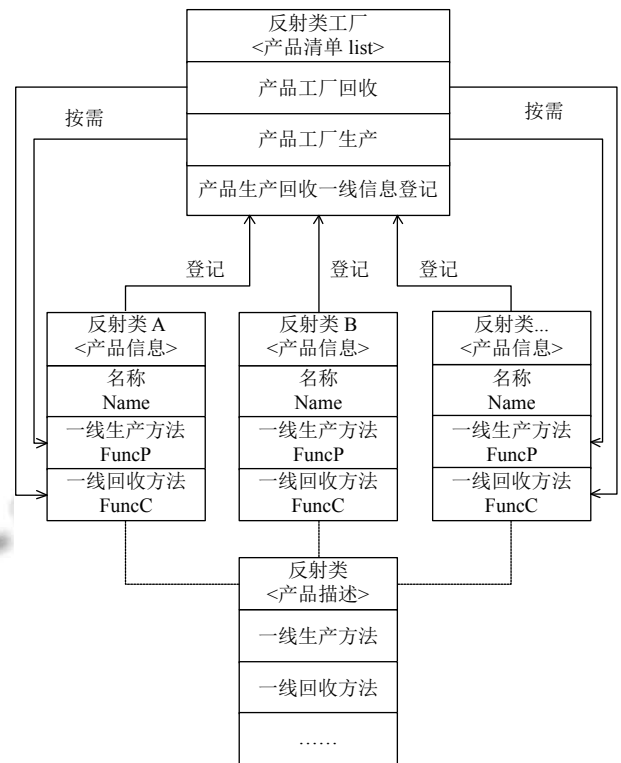


图2 反射类工厂原理示意图

1) 产品生产回收一线信息登记. 反射类工厂统一登记录入厂内所有种类产品的生产回收一线信息(注: 一线同“一线工人”的“一线”词义是指在反射类对象这一产品的生产与回收一线)并生成工厂(可生产回收)产品清单, 其中生产回收一线信息包括产品的名称、一线生产方法、一线回收方法. 工厂产品清单, 可用于支撑后续检索厂内产品并控制相应产品的生产或回收。

2) 产品工厂生产. 收到输入的产品生产要求时, 反射类工厂根据产品名称安排进行生产, 在工厂产品清单中查找到相应产品, 查找到则调用该产品关联的一线生产方法来生产获得产品, 否则告知不是厂内产品无法进行生产. 实现伪代码如下。

输入: 生产产品名称 name, 工厂产品清单 list  
输出: 生产产品 product

```

1. flag←0
2. node←list.begin
3. FOR EACH node!= list.end.next DO
4.   IF node.Name = name THEN
5.     flag←1
6.     break;
7.   ENDIF
8.   node←list.next
    
```

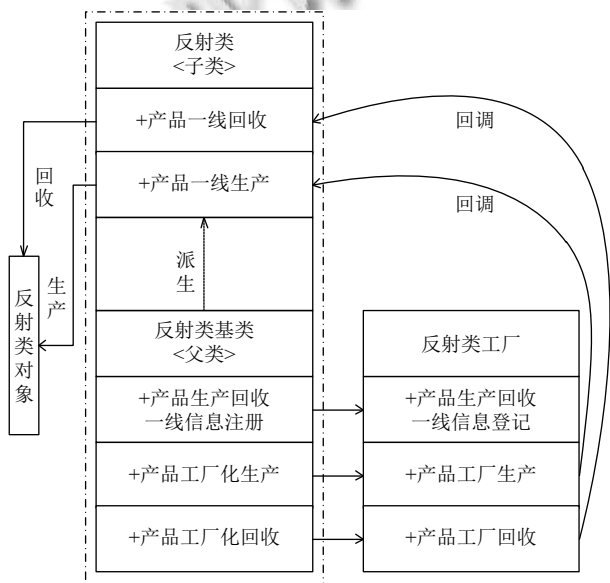
```

9. ENDFOR
10. IF l= flag THEN
11.   product←node. FuncP(name);
12. ELSE
13.   product←NULL;
14. ENDFIF
    
```

3) 产品工厂回收. 原理类同 2). 即收到输入的产品回收要求时, 反射类工厂根据产品名称安排进行回收, 在工厂产品清单中查找到相应产品, 查找到则调用该产品关联的一线回收方法来回收销毁产品, 否则告知不是厂内产品无法进行回收.

(2) 反射类基类

反射类基类即所有反射类的父类, 可派生生成子类即反射类. 它结合运用回调机制, 统一提供反射类工厂产品 (即反射类对象) 的工厂化生产、回收功能, 如图 3 所示.



注: 图中反射类仅表述其部分能力包括继承父类功能、产品一线生产与回收功能.

图 3 反射类基类原理示意图

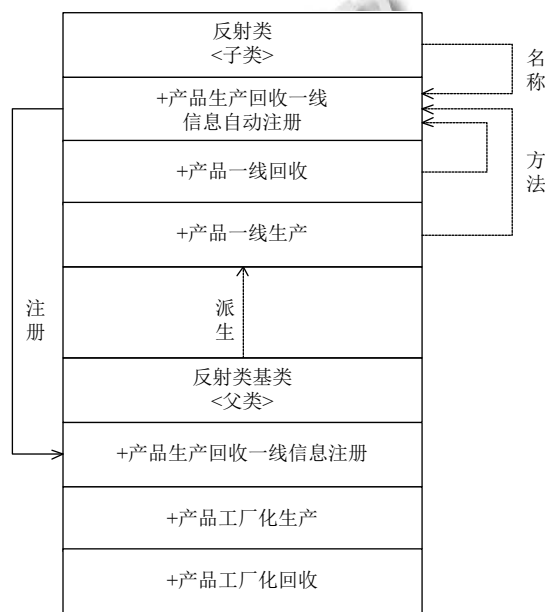
1) 产品生产回收一线信息工厂注册. 反射类基类使用反射类工厂的产品生产回收一线信息登记功能, 完成产品的名称, 及其一线生产与回收方法信息的注册.

2) 产品工厂化生产. 收到输入的产品生产要求时, 反射类基类根据产品名称和产品生产预置参数使用反射类工厂的产品工厂生产功能来完成反射类对象自身的预置参数生产. 注: 该功能实现过程中, 反射类回调了其自身的产品一线生产方法.

3) 产品工厂化回收. 反射类基类收到输入的产品回收要求时, 根据产品名称使用反射类工厂的产品工厂回收功能来完成反射类对象自身的回收. 注: 该功能实现过程中, 反射类回调了其自身的产品一线回收方法.

(3) 反射类通用部件

反射类通用部件可用以构建所有反射类实现的通用组成部分, 它统一提供反射类的产品一线生产与回收并管理有关信息, 如图 4 所示.



注: 图中反射类仅表述其部分能力包括继承父类功能、通用部件功能.

图 4 反射类通用部件原理示意图

1) 产品一线生产. 反射类通过使用自身的产品生产即类对象创建方法来完成产品的一线生产, 运算方法为:

```
obj = new ClassX(paramList)
```

其中, obj为类对象, new为类对象创建的运算符, ClassX为类名称, paramList为类对象生产预置参数.

2) 产品一线回收. 反射类通过自身的产品回收即类对象删除方法来完成产品的一线回收, 运算方法为:

```
delete obj
```

其中, obj为类对象, delete为类对象删除的运算符.

3) 产品生产回收一线信息自动注册. 反射类根据输入的产品名称, 结合自身的产品一线生产与回收方法构成自身的生产回收一线信息, 之后应用自身继承自反射类基类的生产回收一线信息工厂注册功能完成相应信息的注册.

### 2.3 软件构件基类构建

软件构件基类由反射类基类派生而成,除继承反射类基类的所有功能外,还包括软件构件规范化的启停用、功能响应、心跳查询功能接口,如图5(a)所示.



图5 软件构件基类和软件构件类结构

1) 软件构件启用功能接口. 支持各具体构件派生实现自身的启用,使构件开始工作,该接口形式为:

start()

其中, start为构件启用接口执行函数,无输入参数.

2) 软件构件停用功能接口. 支持各具体构件派生实现自身的停用,使构件停止工作,该接口形式为:

stop()

其中, stop为构件停用接口执行函数,无输入参数.

3) 软件构件操作响应功能接口. 支持各具体构件派生实现对外部操作请求自身功能的响应,该接口形式为:

operat(no)

其中, operat为构件操作响应接口执行函数,输入参数包括: no为功能编号.

4) 软件构件心跳查询功能接口. 支持各具体构件派生实现对外部心跳查询请求自身健康状态的响应,即在构件工作正常(即能够响应外部操作请求)的情况下回复对方本构件心跳正常,该接口形式为:

heartAsk(answerFunc)

其中, heartAsk为构件心跳查询接口执行函数,输入参数包括: answerFunc为心跳回复接口函数,其在构件工

作正常的情况下通过该接口告知本构件心跳正常.

### 2.4 软件构件集成管理类构建

软件构件集成管理类能够集成管理软件构件的装卸、启停用、心跳检测、故障修复,如图6所示.

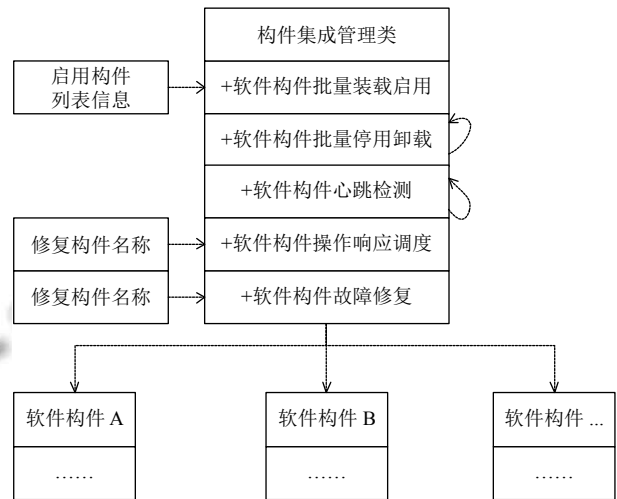


图6 软件构件集成管理类示意图

1) 软件构件批量装载启用. 根据输入的构件信息(含构件名称、产品生产预置参数)列表并更新软件集成的构件信息列表信息,从前往后依次结合运用软件构件的产品工厂化生产、构件启用功能,来打开软件构件实体、生产构件得到相应的操作类对象,之后启用构件使相应的构件开始工作.

2) 软件构件批量停用卸载. 根据自身管理的集成构件信息列表信息,从后往前依次结合运用软件构件的构件停用、产品工厂化回收功能,来停用构件使相应的构件停止工作、回收构件释放相应的操作类对象,之后关闭软件构件实体.

3) 软件构件心跳检测. 根据自身管理的集成构件信息列表信息,从前往后依次运用软件构件的心跳查询功能来查询获取相应构件的心跳,未超时收到心跳则表明构件工作正常,超时收不到心跳则表明构件工作故障.

4) 软件构件操作响应调度. 根据输入要求响应操作的构件名称,在管理的列表中找到有关构件,然后运用软件构件的构件操作响应功能使得相应的构件响应对应功能编号的软件操作.

5) 软件构件故障修复. 根据输入的构件名称,修复相应构件的故障,故障修复手段采用先(使用软件构件的启用功能)重启构件进行修复,构件仍无法恢复工作正常时(使用软件构件的产品工厂化生产功能)重装构

件进行恢复。

### 2.5 软件构件实现

软件构件以类(其名称由构件类标识“CComp”+构件标识组成,构件标识为不超过8个英文字符的非空字符串)的形式呈现,由软件构件基类派生构建,其组成包括反射类通用部件、继承的软件构件基类、以及其它自身的专有功能。其需按照软件构件基类的功能接口规范约定实现自身的启用、停用、操作响应、心跳查询功能,以及按照构件自身的专有功能定义实现有关的功能,如图5(b)所示。

### 2.6 软件按需集成管理

软件通过使用软件构件集成管理类,根据其自身相应的功能构成需求进行构件化集成管理,如图7所示。

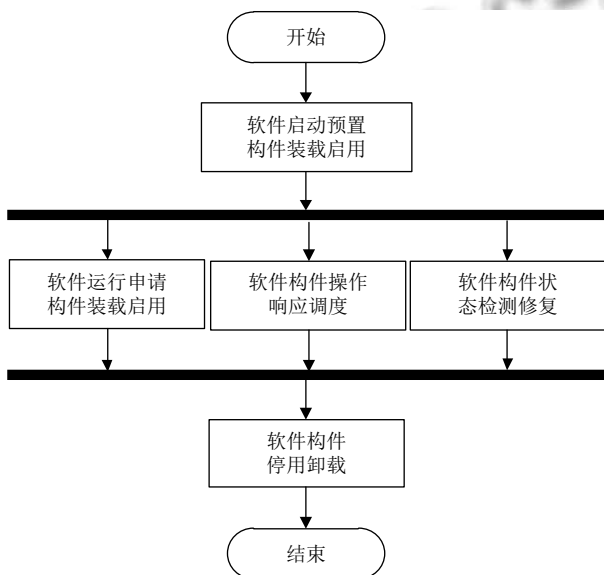


图7 软件按需集成管理活动示意图

1) 软件启动预置构件装载启用。软件启动时,按照软件出厂设定(满足软件功能初始需求)的构件列表配置并使用软件构件集成管理类的构件批量装载启用功能来实现软件预置构件组合的动态集成装载启用。

2) 软件运行申请构件装载启用。软件运行时,可接受新增(满足软件功能临时需求)构件申请并使用软件构件集成管理类的构件批量装载启用功能来实现软件临时需用构件的动态集成装载启用。

3) 软件构件操作响应调度。软件运行时,可接受构件操作响应请求并使用软件构件集成管理类的构件操作响应功能来执行相应编号的软件构件功能操作。

4) 软件构件状态检测修复。软件运行时,可接受构

件心跳检测请求并使用软件构件集成管理类的构件心跳检测功能来检测构件的工作状态,在发现构件工作故障后可以使用构件集成管理类的构件故障修复功能来修复构件故障。

5) 软件构件停用卸载。软件退出时,使用软件构件集成管理类的构件批量停用卸载功能来停用卸载软件集成组装的构件以释放占用的资源。

## 3 方法应用实例

从前文可知,本文的改进软件构件集成方法的实体组成部件为反射类工厂、软件构件基类、软件构件集成管理类,它们不仅是以不同异构(操作系统)环境下都能通用的类和函数的形式来表现,而且可以使用跨平台通用程序语言同时基于运用跨平台通用操作系统接口/操作符如new、delete来实现。故,本文方法具备跨不同异构环境的通用性,且能够通用规范各异构环境下的软件构件开发。进一步,它可通过采用跨平台通用的如C++程序语言来编码进行程序化实现,且其程序源码因方法的可跨异构环境通用性而无需重复编码即可直接复用于Windows、VxWorks、ReWorks、Android、AOS等各类异构环境,而仅需通过各系统配套开发工具上创建工程加载有关源码编译生成相应系统下的软件构件集成服务程序。该程序以动态库的形式对外提供,作为后端服务运行,支撑软件构件的按需集成与功能调度管理。

以Windows系统应用为实例,在系统软件出厂时设定软件默认启用的构件列表配置,其配置以(支持跨平台复用的)XML文件形式提供并按照启动依赖顺序从前到后排序,如图8所示。

```

    <构件列表>
      <构件 标识="Time" 名称="时间构件"/>
      <构件 标识="Position" 名称="位置构件"/>
      <构件 标识="User" 名称="用户构件"/>
      <构件 标识="Notify" 名称="通知构件"/>
      <构件 标识="Navigate" 名称="导航构件"/>
      <构件 标识="Mailbox" 名称="邮箱构件"/>
    </构件列表>
  
```

图8 软件出厂构件列表配置示意图

在软件开发阶段,基于软件构件集成服务程序的构件基类,各软件构件完成自有功能实现并生成构件程序。在软件部署阶段,完成默认启用构件列表配置文件的加载,以及根据构件列表清单完成所有构件的安装。

在软件启动时,软件构件集成服务程序读取默认

启用构件列表配置并据此依次从前到后完成各软件构件的动态集成启用,同时系统软件启动后也可根据临时需求完成新增构件的启用;在系统软件运行过程中,软件构件集成服务程序可以提供软件集成启用的软件构件列表及其运行状态、构件功能操作响应调度、以及控制故障软件构件的修复。同时为用户提供软件构件管理界面,可以实时查看软件集成启用的软件构件列表及其运行状态,并可对软件进行故障修复等管理操作,如图9所示。

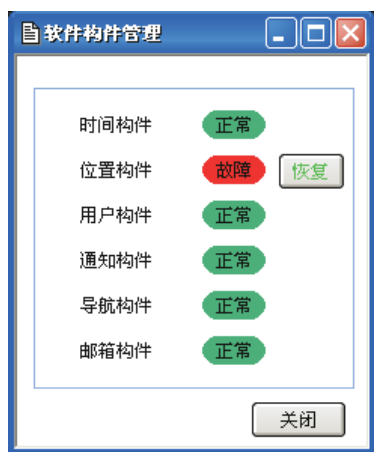


图9 软件构件管理界面示意图

结合上述应用实例和前文的技术方案,可以看出本文方法能够很好地支持跨异构环境可复用,也具备较完备的软件构件集成管理手段。它不仅能够更便捷地复用于各类异构环境上的业务信息系统,而且利于各系统间软件构件的快速跨平台迁移或升级改造,这些都能够为提高信息系统的集成效率、提升软件工程质量水平、以及降低系统的开发维护成本,起着积极的作用。同时,也已成功运用于多型如 VxWorks、ReWorks、Android、WinCE、AOS、Kylin、Windows 等操作系统的陆军业务信息系统中。理论实践均能表明,本文方法适用于电子与信息系统的软件构件化开发集成。

#### 4 结束语

本文立足于为异构环境上的信息系统的提供可复用的跨平台通用软件构件集成手段,论述了类反射式软件构件集成的相关原理、运用机制。同时,给出了面向异构环境可服用的软件构件集成方法,这种方法已成功地在多型嵌入式如单兵业务信息系统中得到应用<sup>[19]</sup>,且也能适用于其它信息系统<sup>[20]</sup>,能够面向陆军领域装

备现状解决按计算等资源环境异构差异需各型适配改造软件集成管理程序的痛点。

#### 参考文献

- 钟莲. 构件技术在物业管理系统中的应用研究 [硕士学位论文]. 北京: 华北电力大学, 2011.
- 梁晓艳. 动态可重组的组件集成框架研究与实现 [硕士学位论文]. 西安: 西安电子科技大学, 2009.
- 蒲海红, 侯秀萍, 赵云峰. 构件集成算法的研究. 计算机技术与发展, 2009, 19(5): 75-78. [doi: 10.3969/j.issn.1673-629X.2009.05.022]
- 倪德强. 基于本体的软件构件描述与检索. 指挥信息系统与技术, 2010, 1(2): 24-28, 38. [doi: 10.3969/j.issn.1674-909X.2010.02.006]
- 常炳国, 王祥宗. 软件构件化生产集成支撑平台研发. 计算机工程与设计, 2011, 32(8): 2712-2715.
- 戴仔强, 金晓雪, 余俊. 基于插件的表现集成技术. 指挥信息系统与技术, 2013, 4(1): 38-42. [doi: 10.3969/j.issn.1674-909X.2013.01.010]
- 黎茂林. 基于构件技术的电子信息系统软件集成平台的设计与实现 [硕士学位论文]. 南京: 南京理工大学, 2013.
- 郭永巧. 构件化软件开发与集成平台的设计与实现 [硕士学位论文]. 西安: 西安电子科技大学, 2013.
- 王霞. 软件复用与软件构件技术. 信息与电脑, 2018, (10): 7-9.
- 柳永齐, 乐勇. 综合电子信息系统面向构件的软件开发与集成技术研究. 电子质量, 2018, (11): 42-44.
- 陈黎明, 高锋. 海军作战指挥软件构件化. 指挥信息系统与技术, 2018, 9(1): 1-7.
- 尤恒. 基于软构件的集成化 CAE 软件开发方法研究. 电子设计工程, 2019, 27(3): 91-94. [doi: 10.3969/j.issn.1674-6236.2019.03.020]
- 武敏. 实例化构件组装平台 [硕士学位论文]. 长春: 吉林大学, 2019.
- 邵刚. 军用软件组件化设计方法研究. 微处理机, 2019, 40(4): 34-37. [doi: 10.3969/j.issn.1002-2279.2019.04.009]
- 吴格格. 基于构件的软件工程技术. 电子技术与软件工程, 2019, (14): 54-55.
- 程知敬, 张晋文, 刘凤. 一种嵌入式信号处理软件构件测试框架. 现代雷达, 2019, 41(6): 82-85.
- 王志坚, 费玉奎, 娄渊清. 软件构件技术及其应用. 第1版. 北京: 科学出版社, 2005.
- 360 百科集成词条. <https://m.baik.e.com/doc/33902-35358.html>. [2020-3-6].
- 徐亮亮, 汤学达, 张媛, 等. 基于 AOS 的军用手持端即时通信软件设计. 指挥信息系统与技术, 2019, 10(5): 86-89, 100.
- 张庆海, 尹瑞, 代杰, 等. 面向网络化指挥控制系统的运维管理软件设计. 指挥信息系统与技术, 2018, 9(4): 68-73.