

组件系统接口匹配识别方案和应用

陈宏君, 王国栋

(南京南瑞继保电气有限公司, 江苏 南京 211102)

摘要:在平台化、组件化的开发背景下,文中提出一种组件系统接口匹配识别方案。该方案首先在版本描述文件中记录组件的当前版本、父版本、兼容标记,并填写所依赖组件的工作版本,之后在集成配置文件中设置各个组件的工作版本。打包工具通过分析组件版本描述文件和集成配置文件,构建单个组件的版本演化树,根据各个组件版本演化树和组件间依赖的工作版本,构建组件系统依赖图,将组件系统的版本匹配拆分为判断各个组件被其他组件所依赖的工作版本是否匹配。如果各个组件在系统内都是匹配的,则整个组件系统是版本匹配的。

关键词:组件系统;版本树;接口匹配;组件依赖

中图分类号:TP39

文献标识码:A

文章编号:1673-629X(2016)02-0124-04

doi:10.3969/j.issn.1673-629X.2016.02.028

Matching Recognition Scheme for Interface of Component System and Its Application

CHEN Hong-jun, WANG Guo-dong

(NR Electric Co., Ltd., Nanjing 211102, China)

Abstract: With the modular philosophy and developing in platform architecture, a matching recognition scheme for interface of component system was presented. Firstly, the scheme records current version, parent version and compatible tokens of component in the version description file as well as the version information of components depended by the current components, then it sets working version of each component in the integrated configuration file. After all that the packaging tool analyzes the version description file and integrated configuration file of each component to construct version evolution tree of every component, and furthermore establish dependency graph of the component system by analyzing the version evolution tree of each component and the working version depended by every components. Finally split the version matching problem of a whole component system into matching of working version of each component depended by other components. And when every component in the system is matched, the whole component system would be version matched.

Key words: component system; version tree; interface matching; component dependency

0 引言

在平台化软件开发和使用周期内,由于面向的应用领域高度复杂,往往存在增强功能、优化性能、修复缺陷等需求,这要求软件系统具有较强的适应能力,以便于开发人员能快速集成。基于组件的软件开发方法可减少软件开发成本和加快开发进度,故组件化的开发得到了广泛应用^[1-5]。复杂、大规模软件体系结构出现网络化、层次化、分布式的发展态势,软件规模扩大化,同构、异构模块存在复杂的交互协同方式,软件的兼容性问题也日益突出^[6]。嵌入式和上位机软件的组件模型在建模过程中可包括 3 类^[7]:源码组件(编译时组件)、二进制代码组件(链接时组件)、可执行代

码组件(运行时组件)。基于组件进行系统组合时,存在一些交互活动的序列不匹配而导致组合行为不兼容的情况^[8-14]。某个组件升级后,应用人员进行集成打包时,存在如下问题或需求:

(1)个别组件升级时,是否其他组件也需配套升级,能否可选升级个别组件;避免所有功能都需完整测试一遍;

(2)如何判断部分升级时,功能接口是匹配兼容的;

(3)某个组件新增的功能,本应用不需要时,是否可保留当前断面版本,不用升级到最新版本;

(4)是否可跨版本升级。

收稿日期:2015-05-10

修回日期:2015-08-13

网络出版时间:2016-01-26

基金项目:国家“863”高技术发展计划项目(2015AA050101)

作者简介:陈宏君(1981-),男,高级工程师,硕士,研究方向为可视化编程软件和嵌入式软件。

网络出版地址:<http://www.cnki.net/kcms/detail/61.1450.TP.20160126.1517.024.html>

在平台化、组件化、模块化架构的开发模式下,希望在解耦发布、可选打包、版本匹配方面能有通用的解决方案,实现组件系统的快速可靠集成。

文中介绍了版本演化树,构建了组件系统依赖图,设计了组件系统接口匹配识别方案。

1 组件概念

组件 (package/component) 是可独立发布的二进制单元。组件是 1 个黑盒子,对外的接口主要是 API 功能和端口,其中 API 是组件对外提供的功能,端口表示组件内部调用外部其他组件功能的函数^[2]。图 1 是一种组件模型。



图 1 组件模型

组件、总线、接口是组件系统的三元组,组件本质是程序,可由界面、函数、数据等组成,它实现某种功能,并通过预定义的接口连接到总线上。总线是一个宿主主控程序,负责组件之间的互操作和通信。接口负责组件的设置、启动、初始化、注销和数据通信等工作。在对组件的接口定义了统一的规范后,系统投入运行时,用户可以根据自己的需要通过接口集成到系统中,或从系统中卸载,而宿主程序和软件框架均不用修改。

2 组件系统接口匹配判据

单个组件基于版本演化信息文件构建版本生成树,按照父节点-子节点层次关系形成 N 叉树,从子节点访问父节点路径需设置兼容、不兼容标志,单个组件内两个版本(A、B)是否兼容的判据是:从版本 A 节点出发,通过递归访问其若干层父节点,可构成 1 条连通路经到版本 B 节点,并且该路径都是可兼容的。

一个系统由 N 个组件构成,各组件之间可能存在版本依赖关系,则整个系统最多构成 $N * (N - 1) / 2$ 条依赖关系。单个组件最多被 $N - 1$ 个组件依赖。则单个组件在系统内是否匹配的判据是:该组件的当前版本和被其他组件依赖的各个版本号之间都是兼容的。如果所有组件在系统内都是匹配的,则整个系统是接口匹配能正常工作的。

3 组件版本演化树

单个组件(模块)存在版本分支,新版本基于某个旧版本升级开发,用 N 叉树结构描述版本演化过程。将新旧版本映射为子-父节点,新旧版本之间存在兼容、不兼容的情况。其三元组为<新版本号,旧版本

号,兼容标记>,见图 2。

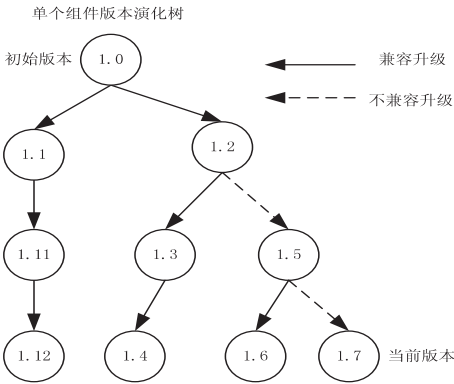


图 2 版本演化树

版本是否兼容,由开发者在发布时标注,如下情况视为兼容:

- (1) 修改 bug,未增加接口、未改变功能行为;
- (2) 新增可选接口函数、可选输入、输出、参数。

如下情况视为不兼容:

- (1) 删除接口、变量;
- (2) 修改接口名、变量名、配置方式;
- (3) 修改了功能行为定义。

其中接口、变量的变化,通过分析对比源文件,词法分析后,进行特定语义分析,提取出函数列表和形参列表,可进行智能校验。当删除原有接口或形参不一致时,置接口不兼容标记,仅新增接口或接口无变化,置接口兼容标记。对于功能函数行为的变化,需人工确定是否兼容,故组件版本节点兼容标志是半自动获取的,是将自动分析获取的接口兼容标记和人工设置的功能行为兼容标记进行与操作后得出组件版本间兼容标记。版本演化树通过读取版本信息文件构建。版本树形结构定义如下:

```
struct VNode {
    String current_version; //当前节点版本
    QString parent_version; //父节点版本
    bool compatible; //是否和父节点兼容
    VNode * parent; //父节点指针
    QList<VNode * > child_list; //第 1 层子节点
};
```

4 组件系统依赖图

一个系统内的组件通常存在版本、功能依赖关系,组件之间能正常协同运行,依赖其他组件有合适匹配的版本(最小工作版本)。将系统的版本匹配拆分为各个组件被要求依赖的版本是否兼容的问题。通过读取组件依赖配置文件,可形成单个组件被要求的工作版本序列,根据版本兼容演化规则,如果单个组件的版本序列是隶属于 1 个可兼容分支,并可演化到当前版本,则进行版本归并,判断为单个组件满足其他组件的

依赖要求。如果所有组件都满足其他组件的依赖要求,则整个系统是版本匹配的。

以图 3 为例,当前系统由 3 个组件构成,分别为 V1.3 版本的 Master, V1.2 版本的 Slave, V1.2 版本的 IEC103。而 V1.3 的 Master 需要的其他组件工作版本为 V1.1 的 Slave。V1.2 版本的 Slave 需要 V1.3 的 Master 配合。V1.2 的 IEC103 依赖 V1.0 的 Slave 和 V1.2 的 Master。

System Packages Version Requirements

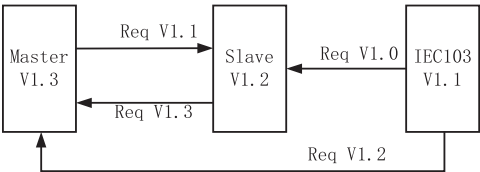


图 3 组件间版本配合依赖关系示意图

- 对每个组件：
- (1) Master 被其他组件依赖的版本序列为：V1.3、V1.2,当前打包版本为 V1.3；
 - (2) Slave 被其他组件依赖的版本序列为：V1.1、V1.0,当前打包版本为 V1.2；
 - (3) IEC103 未被其他组件依赖,当前打包版本为 V1.1；
 - (4) 则版本匹配的问题拆分为 3 个进程各自的版本序列是否是可兼容的;假如 4 个组件的版本演化树如图 4 所示,则可得出该系统的组件之间是版本接口匹配的结论。

```
[ Key = IEC103 Current = 1.1 ]
[ Version Child = 1.1, Parent = 1.0, Compatible = 1 ]
[ DependKey = MASTER, ReqV = 1.3 ] [ DependKey =
SLAVE, ReqV = 1.2 ]
[ Version Child = 1.0, Parent = 1.0, Compatible = 1 ]
```

其中,Key 填写组件名,Current 填写组件当前版本;Child 填写子版本、Parent 填写父版本,Compatible 填写兼容标志;DependKey 填写依赖的外部组件名,ReqV 填写依赖的外部组件工作版本。单个组件发布新版本时,需更新组件版本描述文件,并保留历史版本的描述信息。

集成人员填写组件系统集成配置文件。不同应用人员,所需集成的组件、版本可能不同,通过组件系统集成配置文件,描述该系统所包括的各个组件名字、工作版本。集成配置文件示例如下：

```
Key = MASTER Version = 1.3
Time = 2014-12-06_16:29:59 Crc = 8F6DE121
Key = SLAVE Version = 1.2
Time = 2014-11-11_20:27:19 Crc = 2E9F6B30
Key = IEC103 Version = 1.1
Time = 2014-10-22_10:25:36 Crc = 6A5BC31E
```

其中,组件名 Key,组件工作版本 Version 是关键信息;Time 是组件编译形成时间;CRC 是组件的校验码。

6 具体实现步骤

打包工具通过读取 version.txt 文件、各组件版本描述文件,可构建组件的版本演化树,组件之间依赖版本、组件的工作版本序列,并按照规则得出当前打包的组件之间接口是否匹配的结论。

关键步骤如下：

第一步:定义组件版本描述文件,在版本描述文件中记录组件当前版本、父版本、兼容标记,并保留组件开发历史版本的记录 and 隶属关系,若该组件依赖其他组件,则须填写所依赖组件的工作版本。

第二步:定义组件系统集成配置文件,在该文件中定义所挑选的各个组件名、对应的工作版本。

第三步:分析各个组件版本描述文件,构建单个组件的版本演化树,并判断单个组件的任意 2 个版本是否兼容。用 N 叉树结构描述组件版本演化过程,将新-旧版本映射为子-父节点,其三元组为<新版本号,旧版本号,兼容标记>。从子节点访问父节点路径需设置兼容、不兼容标志,则单个组件内两个版本(A、B)是否匹配的判据是:从版本 A 节点出发,通过递归访问 A 节点的多层父节点,可构成 1 条连通路径到版本 B 节点,并且该路径都是可兼容的。

第四步:分析组件集成配置文件,并根据各个组件

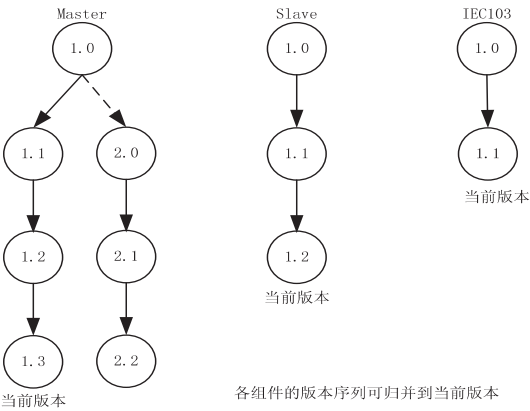


图 4 系统的兼容匹配转换为各个组件的版本序列是否兼容示意图

5 版本描述和集成文本

定义组件版本描述文件格式,组件开发人员按照格式填写对应组件版本信息。组件发布时,开发人员同步提供组件版本描述文件。记录当前版本、父版本、兼容标记,并保留组件开发历史版本的记录 and 隶属关系,如果依赖其他组件,则须填写外部组件的工作版本信息。版本描述文件示例如下：

版本演化树和组件间依赖的工作版本,构建组件系统依赖图,将组件系统的版本匹配拆分为判断各个组件被依赖的工作版本是否匹配。按照第2节的整体设计思路进行判断。

如图5所示,调用版本匹配判断检测工具,组件开发人员定义组件版本描述文件,集成人员定义组件系统配置文件,通过工具导入各个组件版本描述文件和集成配置文件,工具基于第三步、第四步原理进行分析后,得出组件系统版本是否匹配的结论。

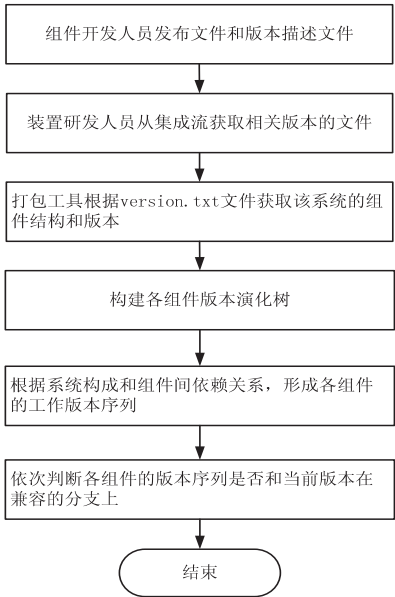


图5 打包时版本匹配的处理流程

7 结束语

基于文中提出的方案,组件集成人员基于版本匹配工具,能快速得出组件系统是否匹配的结论,加快了集成的速度,降低了由于版本不匹配的配置风险。该发明已经在电力系统控制保护平台的嵌入式组件系统、上位机可视化编程配置组件系统的集成时进行了应用,结果表明开发集成效率得到了显著提高。

参考文献:

[1] 陶传奇,李必信, Jerry Gao,等. 基于模型的构件软件修改影响分析[J]. 软件学报,2013,24(5):942-960.

[2] 袁伟民,左 春. 基于样本程序的领域开发平台的研究与实践[J]. 计算机工程与设计,2010,31(18):3979-3982.

[3] 丁 博,王怀民,史殿习. 构造具备自适应能力的软件[J]. 软件学报,2013,24(9):1981-2000.

[4] 周晓锋,马志强,刘馨月. 一种基于组件的软件开发方法[J]. 信息技术与标准化,2005(9):35-38.

[5] 张 驰. 软件组件接口扩展技术研究[J]. 微电子学与计算机,2007,24(8):35-37.

[6] 王 博,白晓颖,贺 飞,等. 可组合嵌入式软件建模与验证技术研究综述[J]. 软件学报,2014,25(2):234-253.

[7] 何鹏飞,何 平,张松阳,等. 组件技术在嵌入式系统中的应用[J]. 计算机系统应用,2014,23(6):220-223.

[8] 郑晓梅,胡晨骏,李 刚,等. 基于 MDE 的 AADL 构件组合兼容的方法[J]. 计算机工程与设计,2014,35(5):1862-1867.

[9] 孙小兵,李必信,陶传奇. 基于 LoCMD 的软件修改分析技术[J]. 软件学报,2012,23(6):1368-1381.

[10] 史浩辉,何 炜. 基于构件的指控软件复用[J]. 计算机技术与发展,2011,21(2):159-161.

[11] Tao C Q, Li B X, Sun X B. A hierarchical model for regression test selection and cost analysis of Java programs[C]//Proc of the Asia Pacific software engineering conf. Sydney: IEEE Computer Society, 2010:290-299.

[12] Lin L, Prowell S J, Poore J H. The impact of requirements changes on specifications and state machines[J]. Journal of Software Practice and Experience, 2009, 39(6):573-610.

[13] Park S, Bae D H. An approach to analyzing the software process change impact using process slicing and simulation[J]. Journal of Systems and Software, 2011, 84(4):528-543.

[14] Hassan M O, Deruelle L, Basson H. A knowledge-based system for change impact analysis on software architecture[C]//Proc of the research challenges in information science. [s. l.]: [s. n.], 2010:545-556.

(上接第68页)

[6] 杨靖宇,张永生,邹晓亮,等. 利用暗原色先验知识实现航空影像快速去雾[J]. 武汉大学学报:信息科学版,2010,35(11):1292-1295.

[7] 李 坤,兰时勇,张建伟,等. 改进的基于暗原色先验的图像去雾算法[J]. 计算机技术与发展,2015,25(2):6-11.

[8] Inampud R B, Purimetla T N, Satyanarayana P G. Contrast degradation for improving quality of an image[J]. Geoscience and Remote Sensing Symposium, 2002, 6:3408-3410.

[9] 杨 帆. 数字图像处理与分析[M]. 北京:北京航空航天大学出版社, 2010.

[10] 高仕龙. 一种基于小波变换和直方图均衡的图像增强方法[J]. 西华大学学报:自然科学版, 2007, 26(3):54-56.

[11] 马云飞,何文章. 基于小波变换的雾天图像增强方法[J]. 计算机应用与软件, 2011, 28(2):71-72.

[12] 吴 强,王新赛,贺 明,等. 一种结合小波分析与直方图的红外图像增强方法[J]. 应用光学, 2011, 32(3):464-467.

[13] 芮义斌,李 鹏,孙锦涛. 一种图像去薄雾方法[J]. 计算机应用, 2006, 26(1):154-156.

[14] 杨 晨. 雾天图像增强算法研究[D]. 南京:南京理工大学, 2007.

[15] 徐贵力,刘小霞,田裕鹏,等. 一种图像清晰度评价方法[J]. 红外与激光工程, 2009, 38(1):180-184.

[16] Tan K, Oakley J P. Enhancement of Color Images in Poor visibility conditions[C]//Proceedings of 2000 international conference on image processing. [s. l.]: [s. n.], 2000:788-791.

组件系统接口匹配识别方案和应用

作者：[陈宏君](#)，[王国栋](#)，[CHEN Hong-jun](#)，[WANG Guo-dong](#)
作者单位：[南京南瑞继保电气有限公司](#)，[江苏 南京](#)，[211102](#)
刊名：[计算机技术与发展](#)[ISTIC](#)
英文刊名：
年，卷(期)：2016 (2)

引用本文格式：[陈宏君](#)，[王国栋](#)，[CHEN Hong-jun](#)，[WANG Guo-dong](#) [组件系统接口匹配识别方案和应用](#) [期刊论文] - [计算机技术与发展](#) 2016 (2)