

# 运行时基于模型的软件动态演化良性建模方法

王 华,李长云,魏秋彦,周玲芳

(湖南工业大学 计算机与通信学院,湖南 株洲 412007)

**摘 要:**针对软件在开放环境下持续运行时用户需求和操作环境不断改变的特点,以及封闭环境假设下演化操作被预先定义的不足,提出了一种基于模型的软件动态演化良性化建模方法。首先运用上下文模型、需求模型、策略模型及体系结构模型等语义丰富的运行时模型,提出了一种基于模型的框架来支持软件的动态演化,以应对开放环境下不可预料的上文事件。如果预定义的演化操作不能解决不确定事件,通过以模型为基础的演化规划来引导模型的良性演化,来维护用户高层次的需求。最后通过简单的原型系统展示了建模方法的可行性和通用性。

**关键词:**上下文感知;模型;动态演化;良性化;建模

中图分类号:TP335

文献标识码:A

文章编号:1673-629X(2015)05-0074-04

doi:10.3969/j.issn.1673-629X.2015.05.018

## A Benign Modeling Method of Software Dynamic Evolution Based on Model in Run Time

WANG Hua,LI Chang-yun,WEI Qiu-yan,ZHOU Ling-fang

(College of Computer and Communication, Hunan University of Technology, Zhuzhou 412007, China)

**Abstract:** Aiming at the characteristics of software users in continuous operation under the open environment requirements and operating environment changing, and closed environment assumption evolution operations are defined in advance, put forward a kind of benign modeling method of software evolution based on dynamic model. First, use of operating context model, demand model, strategy model and system structure model, which are run time model with rich semantics, propose a framework based on model to support the dynamic evolution of software, to cope with unpredictable context event in open environment. If the predefined evolution does not solve the uncertain event, guide the healthy evolution of model by evolutionary programming based on model, to maintain high levels of user demand. Finally, through a simple prototype system shows the feasibility and generalization of the modeling method.

**Key words:** context aware; model; dynamic evolution; benign; modeling

## 0 引 言

目前,社会越来越依赖于持续提供服务的关键软件系统,这类系统部署在开放网络环境中,需要提供 7 \* 24 小时的持续运行服务。为适应复杂环境和需求的变化,软件系统需要进行动态演化,在运行中动态地调整功能和结构,并确保在应对各种改变的同时仍能持续可靠地提供服务<sup>[1]</sup>。Lehman 定律<sup>[2]</sup>论证了软件系统的不断演化性,现实世界的系统可能变得越来越没有价值,也可能变得更加适应环境。软件动态演化并不必然导致软件系统的进化,不恰当的软件动态演

化可能使得软件退化。因此,需要对软件动态演化的良性化程度进行一些研究。

目前关于软件演化通过响应上下文环境和需求改变适应自身的研究,主要集中于借助环境感知领域研究成果,环境显式化表达并且内置于可运行的系统中来推动软件的动态演化。Hussein M 等<sup>[3]</sup>提出了一种基于体系结构的环境感知的软件动态演化方法,使自适应系统中自适应的优点同上下文环境感知系统中建模、处理以及管理上下文环境信息的优点得以结合。Ding B 等<sup>[4]</sup>在 Auxo 框架中显式地维护环境模型和体

收稿日期:2014-07-29

修回日期:2014-10-30

网络出版时间:2015-04-23

**基金项目:**2013 年度科技部科技支撑计划课题(2013BAJ10B14-5);2013 年度国家自然科学基金项目(61350011);2012 年度湖南省自然科学基金(12JJ2036);2014 年湖南省自然科学基金面上课题(14JJ2115);2014 年度湖南省研究生创新基金(CX2014B428);2013 年湖南省教育科研项目(13C030)

**作者简介:**王 华(1986-),男,硕士研究生,CCF 会员,研究方向为软件演化、软件工程;李长云,教授,博士,研究方向为软件演化、可信软件、软件工程。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20150423.1534.005.html>

系结构模型,并基于这些模型和策略连接子的内容来驱动软件的在线变化。张凌浩等<sup>[5]</sup>建立了一种基于进程代数的动态演化模型,通过进程代数对软件的动态演化进行分析。谢仲文等<sup>[6]</sup>以特征为基本部件,以特征组合的方式建立需求模型,并完善了需求模型的规范化来推动软件动态的演化。丁博等<sup>[7]</sup>提出了一种构件模型 ACOE,将软件环境适应能力中的感知、决策、执行等关注点封装成独立的构件和连接子,通过在线重配置使第三方可在必要时通过有选择性的更新来调整适应能力。

上述研究是基于上下文环境、需求的变化、模型驱动来研究软件运行时的动态演化。但是这些建模研究主要关注其中某一单方面的变化引起的软件动态演化,而没有将其结合起来。针对上述问题,文中从上下文环境和用户需求的变化着手,提出一种基于模型的良性动态演化建模方法。通过建立的原型系统来说明该方法在运行时的可行性和通用性。

## 1 良性动态演化研究思路

上下文感知系统的动态演化,可以在运行时通过预定义的适应动作来修改系统的体系结构模型。这种方法在封闭环境假设下的演化效果比较好。然而,在开放的环境中仅仅只有预定义的适应动作是不够的,一些不可预见的上下文事件可能发生。对于这些未知事件产生的不确定性,系统应该具有处理不确定性的方式。

因此,通过系统的良性动态演化旨在使软件更满足需求、更适应环境且演化过程完整和正确。针对这些目标,提出以下策略来处理出现问题的未知上下文事件。策略被用来维护受未知上下文事件负面影响的需求。因此,策略在运行时触发系统的动态演化维护需求。演化的动作则被表示为易于理解的模型。通过模型驱动机制产生的良性动态演化能自动生成演化行为,使系统获得更好的配置。执行动态演化主要通过模型,因为模型与系统之间具有因果联系(如果系统发生变化,它们变化,反之亦然)。

为了使软件系统通过良性动态演化更满足需求、更适应环境以及演化过程的正确和完整,文中提出了一个良性动态演化的框架,如图1所示。通过框架说明模型、系统与上下文环境通过良性动态演化来解决上下文的不确定性。该框架在运行时不断执行不可预期的更改。通过演化一个因果联系的体系结构模型来改变系统架构,支持系统的良性动态演化。

文中提出的框架主要由准备演化阶段、处理演化阶段、执行演化阶段三个阶段所组成。

在准备演化阶段,目的是设置一个可以在运行时

处理未知上下文的模型,这些模型主要可以分为两组:第一组由上下文模型组成;另外一组由运行时上下文感知系统表达需求的需求模型组成。

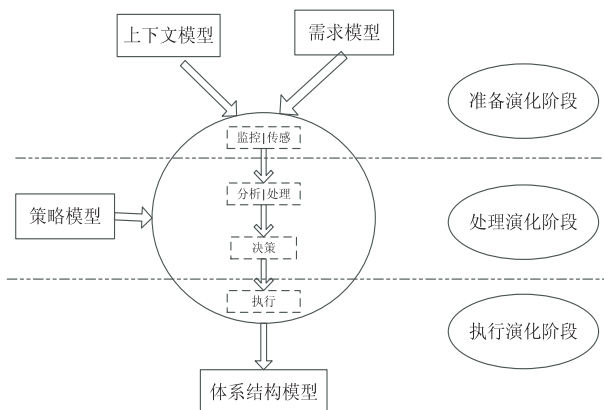


图1 良性动态演化框架

在处理演化阶段,通过前面阶段中上下文监控器持续关注运行时的上下文信息,动态演化模板通过查询上下文监控器所收集到的信息,来决定在需求模型中的需求可能由一个未知的上下文事件所产生的负面影响。为了维护已经受到影响的需求,演化规则根据策略模型产生抽象的演化动作。

在执行演化阶段,重配置引擎根据所产生的演化动作执行来演化相应的体系结构模型。再通过演化相应的体系结构模型来修改系统,使软件系统在运行时达到良性动态演化的目的。

(1)上下文模型。该模型用来支持上下文信息的形式化分析。更具体的说,上下文模型在运行时持续更新上下文信息,来推理何时触发上下文感知的改变。为了检验在预定义的范围内合规的情况,系统分析部分从上下文模型中提取上下文条件作为布尔逻辑表达式。具体来说,上下文模型使用 Web 本体语言 (OWL)<sup>[8]</sup>来支持分析运行时被捕获的上下文信息。特殊的具有用来表示当前上下文状态的数据类型属性。例如,可用性数据类型属性显示一个服务操作目前是否可用。

(2)需求模型。该模型用来表示上下文感知系统在运行时必须维持的需求。文中研究功能性需求的实现程度和运行时质量满足非功能性需求(如安全性和可用性)。这里使用软目标表示系统的需求,因为业务代表核心资产保持系统的服务质量。此外,每个软目标在设计阶段具有定义的优先级。如果一个以上的软目标在运行时已经受到影响,连接到软目标具有最高的优先级的任务被首先执行。此外,每个任务都有一个预定义的优先级。具有最高优先级的任务在运行时首先被选择。

(3)策略模型。当系统没有预定义的适应性动作处理产生问题的上下文事件时,策略被视为处理的最

后方法。高度抽象的策略模型可以用来表示策略功能,策略功能被底层系统触发维护受影响的需求(一个需求可以由多个策略维护)。处理阶段策略模型在必要的应用当前策略激活或禁止当前系统功能。

(4) 体系结构模型。该模型描述了软件体系结构,软件的一个服务组合可以用体系结构模型来表示。基于组件的体系结构系统可以用体系结构描述语言来描述。

2 软件良性动态演化过程

在上节中,提出了一种良性动态演化的模型框架。本节将根据模型框架对良性动态演化的过程进行详细叙述。在良性动态演化过程中(如图 2 所示),当在软件运行时面对未知问题的上下文事件时,整个良性动态演化分为动态演化和动态适应两个大的阶段。

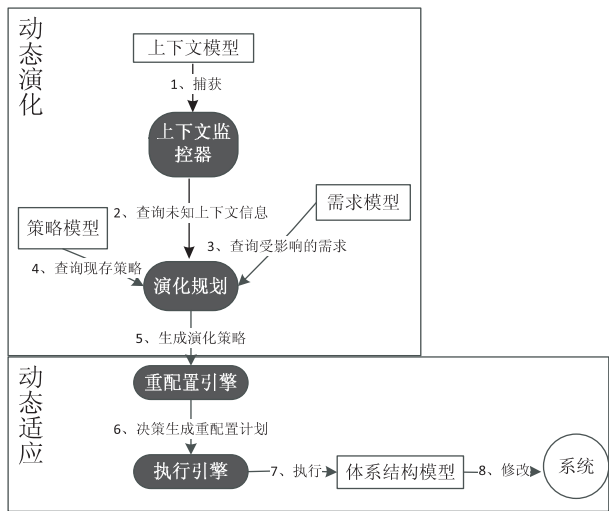


图 2 良性动态演化过程

首先在开放环境中的动态演化阶段,第一步,上下文监控器捕获上下文模型中的重要质量属性的基本指标。而该监控器的监控频率在设计阶段就设计完成。第二步,演化规划不断查询基于上下文监控器所收集的未知问题的上下文事件。第三步,如果演化规划意识出现一个未知上下文事件的情况下,则演化规划会从需求模型中查询受到影响的需求。第四步,根据所

查询到的受到影响的需求,演化规划通过从策略模型中查询一个现存的策略来维护受到影响的需求。第五步,根据触发的策略生成一个演化策略,在演化策略中展现了策略功能的激活或失活特征(如演化的动作)。第六步,根据演化规划生成的演化策略,重配置引擎通过决策生成一个重配置计划。第七步,执行引擎根据生成的重配置计划执行修改了体系结构模型。第八步,体系结构模型通过修改调整底层系统,使得系统更满足需求、更适应上下文环境。

下面详细描述通过演化规划和重配置引擎所执行的动作。

2.1 演化规划

演化规划负责生成演化策略来面对未知问题的上下文事件。演化规划的主要目的是利用现存的策略维护受影响的需求。演化规划的输出是演化策略,重配置引擎根据演化策略来决策功能方面的激活或禁止。

在系统运行时演化规划不断查询通过上下文监控器所查询到的未知上下文事件,而这些上下文事件可能需要系统进行动态演化。触发动态演化最基本的情况是当重配置引擎查询不到任何对于当前上下文事件的上下文条件。在这种情况下,上下文事件被认为是未知的。在演化规划中还需要使用其他的机制,如概率逻辑、模糊逻辑、贝叶斯网络等。而为了处理未知的上下文事件,演化规划按照下面的步骤:

(1) 从需求模型中查询可能受到影响的需求。这一步的目的是查询可能受未知上下文事件影响的需求。这是比较关键的,因为现存的策略与需求因果相关联。为了通过上下文事件找到受影响的需求,文中的演化规划引入了向前链接<sup>[9]</sup>,一种众所周知的关于人工智能的推理方法。该方法评估是在产生的上下文事件行为前提下,而其中的规则的前提是指在设计阶段定义并保存的一个知识库。而在开发的环境中,向前链接的一个关键优势是新的上下文数据能够触发新的推论。

图 3 中叙述了一个基本例子,当未知上下文事件行为(F1)被检测到时,规则R1有一个匹配这个未知

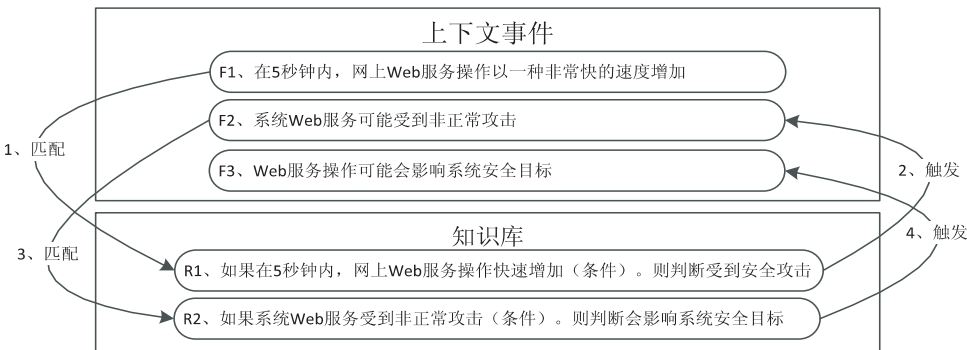


图 3 向前链接事例



上下文的规则条件。然后,通过正向链接方法触发新的上下文事件行为(F2)。该过程继续执行,直到上下文事件行为(F3)被触发。而 F3 则表示通过应用 Web 服务操作来满足未知上下文事件所产生的需求。

(2)从策略模型中查询现存的维护策略。这一步的目的是通过查询现存策略模型中的维护策略来维护已受到未知上下文事件所影响的需求。

(3)产生一个演化策略。这一步的目的是产生一个演化策略,策略中包含演化动作来决定系统的功能在重配置引擎中被激活或禁止。一个演化策略主要由特征功能和特征功能状态(激活或失活)组成。根据系统需求,演化规划触发必要的更改查收演化策略到重配置引擎中。

## 2.2 重配置引擎

重配置引擎决策生成了一个重配置计划,计划中包含了一组重配置动作来适应体系结构模型所表示的底层基础系统。体系结构模型根据重配置计划的执行与修改,对于底层系统进行不断的演化从而与系统需求保持一致。重配置动作在重配置计划中被表述为系统体系结构增量和系统体系结构减量。这些操作以演化策略作为输入,通过增加或者删除模型元素来修改体系结构模型。

## 3 原型系统验证

为了以一种连贯的方式从软件需求到软件运行全过程来执行本建模方法,提出了支持该建模方法的原型系统。首先在准备演化阶段指定要建立模型所需的任务、输出结果、角色和工具。在 Eclipse 过程框架(EPF)<sup>[10]</sup>用于创建一个该建模方法的插件,其作用是其他方法可以引用并使用。在该原型系统中体系结构模型、上下文模型、需求模型、策略模型都被指定使用 XML 元数据(XMI)格式进行数据交换。通过 Eclipse 建模提供的软件基础架构处理框架(EMF)<sup>[11]</sup>在运行时指定和执行对于模型的查询。在文中提出的建模方法中,策略模型符合 MOSKit4SPL<sup>[12]</sup>元模型;体系结构模型符合 SoaML<sup>[13]</sup>元模型,它扩展了 UML2 元模型的支持服务建模。

上述模型利用原型系统来进行动态演化,利用 SALMon<sup>[14]</sup>作为上下文监控器,因为它的组件大多数与技术无关且能够提供服务,使得体系结构按照软件的需求进行修改。在上下文模型中,上下文条件被指定为 SPARQL 协议和 RDF 查询语言(SPARQL)来查询本体。通过使用 SPARQL 查询,实施演化操作来插入新的三元组到本体的 RDF 图,并评价其在上下文模型中的值,以找出是否有预定义的上下文条件已经完成。当一个未知的上下文事件发生,演化规划使用一般规

则推理在 Jena 2<sup>[15]</sup>中来执行向前链接进行推理。另外,演化规划使用 EMF 模型查询(EMFMQ)在执行的过程中激活或者禁用系统功能。执行引擎通过 Swordfish<sup>[16]</sup>来实现。利用 Java API 为 XML Web 服务(JAX-WS)创建 Web 服务并且在 Swordfish 被部署为束。

## 4 结束语

文中提出了一种运行时基于模型的软件动态演化良性化建模方法,来支持软件系统通过良性动态演化更满足需求、更适应环境以及演化过程的正确和完整。通过原型系统的验证来说明该建模方法的适用性和可行性。该方法有几个优点:

(1)通过引导建立上下文模型,面对未知问题的上下文事件能够及时进行良性演化,达到维护满足需求、更适用环境的特点;

(2)对于处理不确定上下文事件的动作被抽象为策略,通过策略在运行时触发系统的动态演化以维护需求;

(3)语义丰富和易于理解的各种模型来促进开发和维护软件需求,达到软件的良性演化;

(4)通过该建模方法进行良性动态演化达到智能的目的,减少工作维护量。

通过总结,有几个未来的工作方向。方向一是在文中建立的良性动态演化模型基础上建立评估指标,来评估良性动态演化;方向二是通过建立的评估指标,提出一种良性动态演化评估方法来具体评估软件良性动态演化的程度。

## 参考文献:

- [1] 王怀民,史佩昌,丁 博,等. 软件服务的在线演化[J]. 计算机学报,2011,34(2):318-328.
- [2] Lehman M M, Pery D E, Ramil J F, et al. Metrics and laws of software evolution—the nineties views[C]//Proceedings of 4th international symposium on software metrics. [s. l.]: IEEE Computer Society Press, 2000:20-32.
- [3] Hussein M, Han J, Colman A, et al. An architecture-based approach to developing context-aware adaptive systems[C]//Proc of 2012 IEEE 19th international conference and workshops on engineering of computer based systems. Novi Sad, Serbia: IEEE, 2012:154-163.
- [4] Ding B, Wang H, Shi D, et al. Taming software adaptability with architecture-centric framework[C]//Proc of IEEE international conference on pervasive computing and communications. Mannheim: IEEE, 2010:145-151.
- [5] 张凌浩,马晓星. 基于进程代数的软件动态演化模型初探[J]. 计算机与数字工程, 2008, 36(10):63-69.
- [6] 谢仲文,李 彤,代 飞,等. 面向软件动态演化的需求建

器的整体分类性能和分类精度都有了明显的提高,且节省了文本的训练和测试时间。

## 5 结束语

文中分析比较了目前常用的几种文本分类方法,指出了 SVM 用于文本分类的优势,并针对现有 SVM 多类文本分类方法中常见的训练或测试过程较长、实际文本空间非线性可分等问题,采用基于二叉树和 Mercer 核的改进 SVM 对多类文本进行识别分类。通过对网络文本进行分类实验后,结果表明,该改进的 SVM 多类分类方法可以较好地解决实际输入文本向量空间呈非线性可分的问题,提高多类文本自动分类精度和泛化能力,缩减文本的训练和测试时间。由于 SVM 是一种基于核的机器学习方法,核函数及相应核参数的选择直接影响到 SVM 的泛化能力,所以在核参数选择方法上还有许多工作要做。

## 参考文献:

- [1] Vapnik V N. 统计学习理论[M]. 许建华,张学工,译. 北京:电子工业出版社,2004.
  - [2] Li Dawei, Xu Lihong, Goodman E D, et al. Integrating a statistical background-foreground extraction algorithm and SVM classifier for pedestrian detection and tracking[J]. Integrated Computer Aided Engineering, 2013, 20(3): 201-216.
  - [3] 边肇祺,张学工. 模式识别[M]. 北京:清华大学出版社, 2000.
  - [4] Tian Yingjie, Shi Yong, Liu Xiaohui. Recent advances on support vector machines research[J]. Technological & Economic
- 
- (上接第 77 页)
- 模及其模型规范化[J]. 计算机科学与探索, 2012, 6(6): 557-576.
- [7] 丁 博,王怀民,史殿习,等. 一种支持软件可信演化的构件模型[J]. 软件学报, 2011, 22(1): 17-27.
  - [8] Wyner A Z. An ontology in OWL for legal case-based reasoning[J]. Artificial Intelligence & Law, 2008, 16(4): 361-387.
  - [9] Labhart J, Rowe M, Matney S, et al. Forward chaining parallel inference[C]//Proceedings of the 2nd IEEE symposium on parallel and distributed processing. Dallas, TX: IEEE, 1990: 455-462.
  - [10] Kerner S M. Eclipse takes 'best' shot at framework[R]. [s. l.]: [s. n.], 2006.
  - [11] Arendt T, Taentzer G. A tool environment for quality assurance based on the eclipse modeling framework[J]. Automated Software Engineering, 2013, 20(2): 141-184.
  - [12] MOSKitt4SPL is a free open-source tool which gives support

- Development of Economy, 2012, 18(1): 5-33.
- [5] Manikandan J, Venkataramani B. Study and evaluation of a multi-class SVM classifier using diminishing learning technique[J]. Neurocomputing, 2010, 73(10-12): 1676-1685.
  - [6] 段 莹. 支持向量机在文本分类中的应用[J]. 计算机与数字工程, 2012, 40(7): 87-88.
  - [7] 崔建明,刘建明,廖周宇. 基于 SVM 算法的文本分类技术研究[J]. 计算机仿真, 2013, 30(2): 299-302.
  - [8] Maksim L, Matthias H, Bernt S. Learning using privileged information; SVM+ and weighted SVM[J]. Neural Networks, 2014, 53(5): 95-108.
  - [9] 黄振龙,郑 骏,胡文心. 基于类间可分性 DAG-SVM 的文本分类[J]. 华东师范大学学报: 自然科学版, 2013(3): 209-218.
  - [10] 许世明,武 波,马 翠,等. 一种基于预分类的高效 SVM 中文网页分类器[J]. 计算机工程与应用, 2010, 46(1): 125-128.
  - [11] Rojo-Álvarez J L, Martine-Ramón M, Muñoz-Mari J, et al. A unified SVM framework for signal estimation[J]. Digital Signal Processing, 2014, 26(3): 1-20.
  - [12] 姜 鹤,陈丽亚. SVM 文本分类中一种新的特征提取方法[J]. 计算机技术与发展, 2010, 20(3): 17-19.
  - [13] 郑 伟,吕建新,张建伟. 文本分类中特征预抽取方法研究[J]. 情报科学, 2011, 29(1): 86-88.
  - [14] Chang C C, Lin C J. LIBSVM: A library for support vector machines[EB/OL]. 2012-04-04. <http://www.csie.ntu.edu.tw/~cjlin/papers/libsvm.pdf>.
  - [15] Hsu C W, Chang C C, Lin C J. A practical guide to support vector classification[EB/OL]. 2010-04-15. <http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf>.
- 
- for modeling Software Product Lines (SPL) [EB/OL]. 2012. <http://www.pros.upv.es/m4spl>.
- [13] Ali N, Nellipaiappan R, Chandran R, et al. Model driven support for the service oriented architecture modeling language [C]//Proc of international conference on software engineering. [s. l.]: [s. n.], 2010: 8-14.
  - [14] Ameller D, Franch X. Service level agreement monitor (SA-LMon) [C]//Proceedings of the seventh international conference on composition-based software systems. Madrid: IEEE, 2008: 224-227.
  - [15] McBride B. Jena: a semantic web toolkit[J]. IEEE Internet Computing, 2002, 6(6): 55-59.
  - [16] Swordfish is a standalone eclipse application supporting deployment of swordfish modules from the eclipse IDE [EB/OL]. 2014. [http://archive.eclipse.org/archived\\_projects/swordfish](http://archive.eclipse.org/archived_projects/swordfish).

运行时基于模型的软件动态演化良性建模方法

作者：[王华](#)，[李长云](#)，[魏秋彦](#)，[周玲芳](#)，[WANG Hua](#)，[LI Chang-yun](#)，[WEI Qiu-yan](#)，[ZHOU Ling-fang](#)

作者单位：[湖南工业大学 计算机与通信学院, 湖南 株洲, 412007](#)

刊名：[计算机技术与发展](#)

英文刊名：[Computer Technology and Development](#)

年，卷(期)：[2015 \(5\)](#)

引用本文格式：[王华](#).[李长云](#).[魏秋彦](#).[周玲芳](#).[WANG Hua](#).[LI Chang-yun](#).[WEI Qiu-yan](#).[ZHOU Ling-fang](#) [运行时基于模型的软件动态演化良性建模方法](#)[期刊论文]-[计算机技术与发展](#) 2015 (5)