

<<软件动态演化技术>>

图书基本信息

书名：<<软件动态演化技术>>

13位ISBN编号：9787301129890

10位ISBN编号：7301129890

出版时间：2007-11

出版时间：北京大学

作者：李长云,何频捷,李玉龙

页数：290

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<软件动态演化技术>>

内容概要

为了适应Internet开放环境和用户需求的不断变化，软件系统需要不断调整自身。

软件动态演化技术是满足这一变化的有效手段，也是自治计算、网格计算、自适应软件和网构软件的核心技术。

本书是国内外第一本对软件动态演化技术进行系统阐述的著作。

作者结合多年研究和实践的经验，从开放系统发展动力到动态演化技术产生，从动态演化技术基本原理到动态演化的形态和粒度，从动态配置技术到基于反射的动态演化、动态演化的基础设施，从设计可动态演化的软件系统到动态演化技术的应用以及未来发展趋势，都进行了系统的论述。

本书最后部分介绍了作者提出的软件动态演化模型SASM及其支持工具和平台，望起到抛砖引玉的作用。

本书内容全面、叙述清楚，注意一些最新的协议、规范及学术界、工业界研究进展，同时还非常注重实用性。

本书既适用于本科高年级和研究生的教学，也可供工程技术人员自学参考之用。

<<软件动态演化技术>>

书籍目录

第1章 概述	1.1 软件演化的基本概念	1.1.1 软件演化、软件维护与软件复用	1.1.2 软件演化的分类
1.2 设计时演化	1.2.1 设计模式对设计时演化的支持	1.2.2 构件技术对设计时演化的支持	1.2.3 框架技术对设计时演化的支持
1.3 装载时演化	1.4 动态演化概念	1.4.1 动态演化、动态配置和软件的演化性	1.4.1 动态演化分类
1.5 软件动态演化技术的重要性	1.5.1 Internet需要软件动态演化	1.5.2 动态演化性是网构软件的基本特征	1.5.3 追求动态演化能力是自治计算的目的
1.5.4 动态演化技术是网格计算的基础	第2章 动态演化基础	2.1 基本原理	2.1.1 动态演化过程
2.1.2 语言、模型和平台	2.1.3 动态演化要解决的关键问题	2.2 系统一致性	2.2.1 系统一致性分类
2.2.2 行为一致性	2.2.3 构件状态一致性	2.2.4 应用状态一致性	2.2.5 引用一致性
2.3 状态迁移方法	2.3.1 状态检测	2.3.2 状态迁移方法分类	2.3.3 一种构件间状态迁移的元模型驱动方法
第3章 设计可动态演化的软件系统	3.1 构造性和演化性	3.2 动态需求	3.2.1 具有动态性的需求和需求的动态性
3.2.2 需求的动态变化性	3.2.3 具有动态性的需求	3.3 应用设计模式	3.3.1 设计模式的概念和分类
3.3.2 支持动态演化的设计模式	3.3.3 设计模式的应用	3.4 应用框架	3.4.1 框架的概念和分类
3.4.2 支持动态演化的框架	3.4.3 框架的应用	3.5 应用软件体系结构风格	3.5.1 体系结构风格概念和分类
3.5.2 支持动态演化的体系结构风格	3.5.3 体系结构风格的应用	3.6 AOP技术	3.6.1 AOP技术简介
3.6.2 动态AOP	3.6.3 AOP技术在Java平台中的应用	第4章 动态演化的粒度	4.1 函数层次的动态演化
4.1.1 DLL简介	4.1.2 调用方式	4.1.3 重新编译问题及解决方案	4.1.4 小结
4.2 类/对象层次的动态演化	4.2.1 JAVA的动态性	4.2.2 隐式加载和显式加载	4.2.3 自定义类加载机制
4.2.4 类加载器的阶层体系	4.2.5 类的动态替换	4.2.6 小结	4.3 构件层次的动态演化
4.3.1 构件和基于构件的软件工程	4.3.2 当前主要的构件标准规范	4.3.3 构件的动态配置	4.3.4 总结
4.4 动态软件体系结构	4.4.1 体系结构概念	4.4.2 演化与体系结构	4.4.3 动态软件体系结构的描述
4.4.4 动态软件体系结构的实现	4.5 动态 workflow	4.5.1 workflow 技术简介	4.5.2 动态 workflow 概述
4.5.3 动态 workflow 的特征及分类	4.5.4 动态修改的策略或处理	4.5.5 应用示例	第5章 动态配置技术
5.1 动态配置系统体系结构	5.2 动态配置方法的分类	5.3 避免性动态配置方法	5.3.1 Jeff方法
5.3.2 Warren方法	5.3.3 其他方法	5.3.4 避免性动态配置方法中存在的不足	5.4 动态配置算法
5.4.1 构件删除算法	5.4.2 构件添加算法	5.4.3 构件替换算法	5.4.4 构件迁移算法
5.4.5 连接建立算法	5.4.6 连接删除算法	5.4.7 连接重定向算法	5.4.8 构件属性设置算法
第6章 基于反射的动态演化	6.1 反射	6.1.1 背景、概念和特征	6.1.2 反射的分类
6.2 反射系统	6.2.1 反射系统的概念	6.2.2 面向对象的反射系统	6.2.3 反射模型
6.3 反射和演化	6.4 反射中间件	6.4.1 背景和概念	6.4.2 几个典型的反射中间件
6.4.3 中间件中的反射层	6.4.4 反射层的编程模型	6.4.5 利用反射层实现服务定制	6.5 基于反射理论的动态配置模型
第7章 动态演化的基础设施	7.1 COM构件的演化机制	7.1.1 概述	7.1.2 平台设计
7.1.3 ProBase平台引擎的设计	7.1.4 业务构件交互问题	7.1.5 ProBase优点总结	7.2 CORBA构件的演化机制
7.2.1 概述	7.2.2 反射式动态配置模型RDRM	7.2.3 动态配置系统反射体系	7.2.4 RDRM模型中的要素活跃性分析
7.2.5 RDRM模型在CCM平台上的映射	7.2.6 StarDRP的实现	7.2.7 StarDRP体系结构	7.2.8 小结
7.3 J2EE平台的演化机制	7.3.1 构件管理框架	7.3.2 J2EE动态演化支撑平台	7.3.3 小结
7.4 Web Services和SOA	7.4.1 Web Services技术	7.4.2 SOA基础	7.4.3 SOA与Web Services的联系
7.4.4 Web Services的动态组合	7.5 多Agent系统	7.5.1 多Agent系统简介	7.5.2 多Agent系统的体系结构
7.5.3 多Agent系统的动态性分析	7.5.4 Web Agent	第8章 与动态演化技术相关的应用	8.1 自治计算
8.1.1 自治计算的概念	8.1.2 自治计算的特征	8.1.3 动态演化在自治计算中的应用	8.2 网格计算
8.2.1 网格计算的概念	8.2.2 网格计算的体系结构	8.2.3 网格软件构件	8.2.4 网格服务集成
8.3 普适计算	8.3.1 普适计算的概念	8.3.2 普适计算层次化模型	8.3.3 普适计算的关键技术
8.3.3 动态演化在普适计算中的应用	8.4 自适应中间件	8.4.1 自适应中间件的概念	8.4.2 自适应中间件的分类
8.4.3 自适应中间件的支撑方法	第9章 支持动态演化的模型SASM	9.1 引言	9.2 D-ADL语言
9.2.1 D-ADL设计原则	9.2.2 高阶多型开演算简介	9.2.3 D-ADL的语法规约	

<<软件动态演化技术>>

和形式语义 9.2.4 D-ADL对系统联机演化和SA求精的形式化支持 9.2.5 D-ADL和其他相关工作的比较 9.3 SASM模型 9.3.1 相关研究 9.3.2 基本原理 9.3.3 SASM框架 9.3.4 SASM中的反射机制 9.3.5 SASM系统开发 9.4 SASM动态演化方法 9.4.1 简单的系统演化 9.4.2 由RSAS变更引起的动态演化 9.5 小结第10章 SASM支撑平台和工具 10.1 引言 10.2 支撑平台的总体架构设计 10.3 运行和监控的关键技术 10.3.1 运行信息跟踪器的机理分析 10.3.2 元连接件引擎的设计 10.4 动态演化管理 10.4.1 动态演化过程中的平台支持 10.4.2 运行状态维持机制 10.5 支撑平台的一个原型实现 10.5.1 原型系统的开发环境 10.5.2 体系结构元素的表示 10.5.3 原型系统的设计与实现 10.5.4 对原型环境中物理构件间的通信测试参考文献

<<软件动态演化技术>>

章节摘录

第1章 概述 1.1 软件演化的基本概念 “变化”是现实世界永恒的主题，只有“变化”才能发展。

Lehman认为，现实世界的系统要么变得越来越没有价值，要么进行持续不断的演化变化以适应环境的变化。

软件是对现实世界中问题空间与解空间的具体描述，是客观事物的一种反映。

现实世界是不断演化的，因此，演化性是软件的基本属性。

特别是在Internet成为主流软件运行环境之后，网络的开放性和动态性使得客户需求与硬件资源更加频繁地变化，导致软件的演化性和复杂性进一步增强。

1.1.1 软件演化、软件维护与软件复用 软件演化（Software Evolution）指在软件系统的生命周内软件维护和软件更新的行为和过程。

在现代软件系统的生命周内，演化是一项贯穿始终的活动，系统需求改变、功能实现增强、新功能加入、软件体系结构改变、软件缺陷修复、运行环境改变无不均要求软件系统具有较强的演化能力，能够快速适应改变，减少软件维护的代价。

软件演化研究致力于寻找规则、识别模式。

这种规则和模式，主要用于管理那些由应用程序的编程人员或者是维护人员所作的变更。

这意味着软件演化研究与导致系统不一致的变化有关。

软件演化是一个模糊而范围广泛的概念，普遍的理解认为演化就是修改软件，而且意味着需要一个新的开发周期，包括新的测试和生成阶段，这样对于资源消耗来说是很危险的。

尤其当调试者不是软件的开发人员的时候，就显得更加复杂了。

我们认为软件演化应致力于尽可能限制那些可能被影响的变化本身的性质。

从软件演化的概念来看，软件演化和软件维护有着密切联系，但二者又有本质区别。

软件维护是对现有的已交付的软件系统进行修改，使得目标系统能够完成新的功能，或是在新的环境下完成同样的功能，主要是指在软件维护期的修改活动。

而软件演化则是着眼于软件的整个生命周期，从系统功能行为的角度来观察系统的变化，这种变化是软件的一种向前的发展过程，主要体现在软件功能的不断完善。

在软件维护期，通过具体的维护活动可以使系统不断向前演化。

因此，软件维护和软件演化可以归结为这样一种关系：前者是后者特定阶段的活动，并且前者直接是后者的组成部分。

<<软件动态演化技术>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>