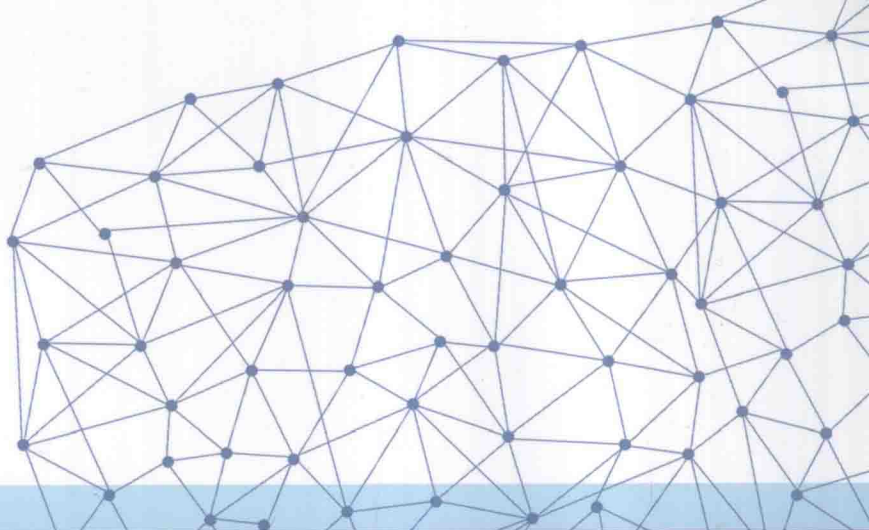
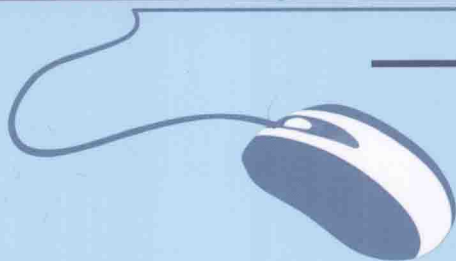




网构软件演化技术

——面向多层架构



WANGGOU RUANJIAN YANHUA JISHU — MIANXIANG DUOCENG JIAGOU

韦正现 宋敏 周连科 著 <<

HEUP 哈尔滨工程大学出版社

网构软件演化技术

——面向多层架构

韦正现 宋敏 周连科 著

 哈尔滨工程大学出版社

内容简介

网构软件是21世纪计算机软件的发展方向,要求在开放、动态、难控的环境下,为用户提供7×24、不断演化更新、智能化的服务,需要相应的演化机制方法支撑。本书系统地介绍了面向多层架构的网构软件演化技术基本概念和新的特征,从语义关系角度,阐述了网构软件静态演化特征与波及效应,从数据流和控制流角度,探讨了网构软件动态演化错误和运行实例动态可迁移准则,并展望了网构软件演化技术发展趋势。

本书可为计算机软工工程技术领域科技人员提供参考,也可供高等院校计算机技术、软件技术等专业的教师及研究生阅读参考。

图书在版编目(CIP)数据

网构软件演化技术:面向多层架构/韦正现,宋敏,
周连科著. —哈尔滨:哈尔滨工程大学出版社,2016.1

ISBN 978-7-5661-1211-8

I. ①网… II. ①韦… ②宋… ③周… III. ①软件
设计-研究 IV. ①TP311.5

中国版本图书馆 CIP 数据核字(2016)第 010258 号

出版发行 哈尔滨工程大学出版社
社 址 哈尔滨市南岗区东大直街 124 号
邮政编码 150001
发行电话 0451-82519328
传 真 0451-82519699
经 销 新华书店
印 刷 哈尔滨市石桥印务有限公司
开 本 787 mm × 960 mm 1/16
印 张 9
字 数 154 千字
版 次 2016 年 1 月第 1 版
印 次 2016 年 1 月第 1 次印刷
定 价 25.00 元

<http://www.hrbeupress.com>

E-mail: heupress@hrbeu.edu.cn

前言

“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,演化是软件的基本属性。目前网构软件已经成为 Internet 上的主流软件系统,网络的开放性和动态性使得网构软件的演化性出现了新的特征和挑战。对网构软件演化技术进行研究与分析,能为我国在未来建立面向 Internet 的软件产业打下坚实的基础,能为我国软件产业的跨越式发展提供技术支持。

面向多层架构的网构软件是 Internet 上来自不同厂商的各种软件实体(构件、服务或 Agent 等,统称为构件)以开放、自主的方式存在于各个节点,在开放的环境下以各种协同方式实现跨网络的互联、互通和协作的软件联盟。网构软件在多层架构支持下,要求在开放、动态、难控的环境下,为用户提供 7×24 、不断演化更新、智能化的服务,需要相应的演化技术和机制支撑。网构软件的演化技术分为静态演化和动态演化。静态演化研究要求从网构软件的体系结构(Software Architecture, SA)层面出发,确定发生变化的构件(演化源)对体系结构的影响范围;而动态演化则需要确定网构软件实体的运行实例是否可以从当前的流程模式(源模式)迁移到新的流程模式下(目标模式)继续执行。当前在静态演化研究中,发生变化的构件(演化源)对体系结构的影响范围缺乏一个定量的分析方法;而动态演化研究是将数据流依附在控制流上,不能满足网构软件核心理论之一——软件协同分离化的一个方面——数据流与控制流分离的要求。

针对上述要求与问题,我们撰写了本书。书中结合作者长期参与网构软件技术研究,在参考国内外相关文献的基础上,融进了作



者大量的最新研究成果,既注重理论与方法的完整性,又注重理论与方法的工程实用性。本书共分为7章,各章节内容介绍如下。

第1章 绪论。介绍了网构软件演化技术的研究目的和意义、网构软件静态演化和动态演化的核心问题,论述了本书的主要组织结构和研究内容。这为后面展开进一步的研究打下了基础。

第2章 基本概念。首先,通过与传统软件系统的对比,介绍了面向多层架构的网构软件基本概念、技术特点以及在软件工程方法上带来的变化和挑战;其次,介绍了软件演化的基本概念,重点论述了软件静态演化和动态演化的技术特点;再次,论述了网构软件演化技术新特征和要求;最后,对网构软件演化技术的国内外现状进行了阐述。

第3章 网构软件静态演化特征。首先,阐述了网构软件体系结构中构件本身及构件之间语义关系及其特征,基于语义协议关系,分析构件内部和构件之间语义的时序逻辑性,采用语义协议关系项来表示构件内部和构件之间语义的时序逻辑性,探讨了网构软件体系结构的语义关系矩阵和语义关系链矩阵,并对它们的性质进行了分析;在此基础上,分析网构软件中构件增加、修改、删除、合并和拆分等演化操作特性,形成了体系结构演化操作及其影响的定量的衡量指标,为量化评价衡量静态演化波及效应提供了技术基础。

第4章 基于语义关系的网构软件静态演化波及效应。首先,根据网构软件体系结构语义关系特征以及静态演化操作特性,论述了构件端口语义和端口之间通过方法调用组合形成的网构软件语义的形式化描述,以及面向网构软件静态演化特性的体系结构形式化模型;然后,基于网构软件体系结构的语义关系矩阵及其相关性质,分析了用于量化分析衡量静态演化操作影响范围的语义关系链波及效应指数和语义关系构件波及效应指数;在此基础上,根据构件之间语义协议关系的强弱程度,阐述了构件增加、修改、删除、合并和拆分等演化操作对体系结构的影响关系和程度,探讨了确定不同演化操作的语义关系链波及效应和语义关系构件波及效应的

算法。

第5章 网构软件动态演化错误。首先,基于着色 Petri 网描述了一种面向数据流和控制流的网构软件模型,将数据流显式地引入网构软件建模过程中;其次,从静态关系和动态运行两个方面分析网构软件服务实施过程中数据依赖关系及其特性,体现数据流和控制流在网构软件动态适应性和演化性中并重的特点;在此基础上,重点分析了网构软件模型数据依赖关系的性质,强调在动态演化过程中必须保持的数据依赖关系;然后,以面向数据流和控制流的网构软件模型为手段,分析了网构软件动态演化操作方式和特征,探讨了网构软件动态演化过程中数据流约束关系、数据流和控制流交叉依赖关系的变化,导致数据流和控制流方面可能出现的动态演化错误特征。

第6章 网构软件运行实例动态迁移准则。首先,以面向数据流和控制流的网构软件模型为基础,针对实现数据流和控制流分离以后的网构软件模型中数据依赖关系特性以及动态演化可能产生的错误,论述了面向数据流的网构软件运行实例的动态可迁移性准则,不但考虑了源模式和目标模式之间的数据依赖关系,还考虑了源模式和目标模式之间参数非重复性复现的情况,同时根据可迁移性准则要求设计关于数据依赖关系的生成算法;然后,根据网构软件数据流和控制流分离后,它们之间的交叉作用特性,探讨了网构软件运行实例关于数据流/控制流交叉依赖关系的动态可迁移性准则,并证明该迁移性准则确保发生演化的实例的目标状态是有效的,从而刻画出完整的网构软件运行实例的动态迁移约束特性。

第7章 总结与展望。本章内容是对本书工作的一个概括和总结,以及对一些关于进一步工作研究方向的建议和思考。

本书由中国船舶工业系统工程研究院的韦正现同志总体策划和组织,并撰写了第1章、第5章和第6章,北京外国语大学的宋敏同志撰写了第3章和第4章,哈尔滨工程大学的周连科老师撰写了第2章,并参与第5章和第6章的编写。同时在编写过程中得到张哲、徐国忠、印桂生、董红斌、王念滨、王红宾等同志的支持与帮助。



本书出版得到了国家自然科学基金项目(61502037, 40746029, 60973028, 61170209, 61102105)、基础科研项目(JCKY2016206B001, JCKY2015206C002)、技术基础研究项目(JSJC2013206C512)、中央高校基本科研业务费专项资金资助项目(2014JJ009, HEUCF100605)、哈尔滨市自然科学基金资助项目(2009RFQXG026)的联合资助,在撰写本书过程中,参考了相关文献中的研究成果,在此一并表示感谢。

网构软件演化技术是一个新的研究方向,其理论及应用均有大量问题有待进一步研究。我们期望本书能够为读者的研究提供一定的启发,对本领域的发展起到一定的促进作用。由于作者学识水平有限、经验不足,书中尚有不妥之处,敬请同行专家和读者批评指正。

著 者

2016年1月

目 录

第1章 绪论	1
1.1 网构软件演化技术研究的目的和意义	1
1.2 网构软件演化技术	4
1.3 主要工作与组织结构	7
第2章 基本概念	12
2.1 面向多层架构的网构软件	12
2.2 软件演化基本概念	16
2.3 网构软件演化特征	30
2.4 网构软件演化技术研究进展	33
第3章 网构软件静态演化特性	40
3.1 引言	40
3.2 网构软件体系结构语义关系特征	41
3.3 SA 语义关系矩阵及语义关系链矩阵	47
3.4 SA 语义演化操作与分析	51
3.5 实例与分析	56
3.6 小结	62
第4章 基于语义关系的网构软件演化波及效应	64
4.1 网构软件语义关系链矩阵构建方法	64
4.2 网构软件语义关系波及效应算法	70
4.3 实例与分析	73
4.4 小结	77
第5章 网构软件动态演化错误	79
5.1 问题的提出	79
5.2 预备知识	82
5.3 网构软件模型	85



5.4	IW_CPN 数据依赖关系分析	89
5.5	面向数据流的网构软件动态演化错误分析	93
5.6	小结	100
第 6 章	网构软件运行实例动态迁移准则	102
6.1	运行实例动态迁移	102
6.2	面向数据流的网构软件运行实例可迁移准则	106
6.3	面向数据流/控制流的运行实例可迁移性准则	113
6.4	实验与分析	115
6.5	小结	122
第 7 章	总结与展望	123
参考文献		126

第1章 绪 论

1.1 网构软件演化技术研究的 目的和意义

随着 Internet 技术的不断发展,基于 Internet 的网络应用也在日新月异地变化,使计算机软件开发、部署、运行和维护的环境开始从封闭的、静态的、可控环境逐步向开放的、动态的和多变的 Internet 环境转变,软件系统也开始呈现具有自主适应、动态协同、在线演化、连续反应等形态特征,从而出现了一种新的软件形态——网构软件(Internetware)。网构软件是 Internet 上来自不同厂商的各种软件实体(构件、服务或 Agent 等,统称为构件)以开放、自主的方式存在于各个节点,在开放的环境下以各种协同方式实现跨网络的互联、互通和协作的软件联盟。它具有自主适应性、协同性、反应性、演化性、多态性等形态特征,为用户提供 7×24 、不断演化更新和智能化的服务,同时能感知到外部环境的动态变化,并且根据功能指标、性能指标和可靠性指标等进行在线动态演化。杨芙清教授在 2005 年指出,建立一套完整的网构软件理论、方法、技术与平台,一方面为 21 世纪计算机软件的发展构造理论基础;另一方面,为我国在未来 5~10 年建立面向 Internet 的软件产业打下坚实的基础,为我国软件产业的跨越式发展提供核心技术支持。面对网构软件这种新型的软件形态,传统的软件理论、方法、技术和平台面临着一系列挑战,同时这种挑战为我们研究具有前瞻跨度大、原始创新性强的软件理论、方法和技术提供了难得的机遇^[1,2]。

网构软件的理论、方法和技术要求在开放、动态、难控的环境下,为软件系统的开发、运行和维护提供直接、自然和有效的支持。因为网构软件概念框架、逻辑内涵等和相对封闭的、静态的、流程固定的经典软件系统存在很大差异,因此,相对于经典的软件系统,网构软件在理论、模型、方法、技术上必然存



在一系列新的挑战^[2]:

(1) 基本理念开放化 在开放的网络环境下,如何全面完整地认识基础平台、应用需求以及软件行为等基本特征。

(2) 软件实体主体化 从开放的软件开发方法与系统运行方式上,如何使软件实体具有并保持内容自包含、结构自独立与实体自适应等特性。

(3) 软件协同分离化 从软件结构模型中软件实体之间的协同角度,如何将协同机制从软件实体的计算逻辑中分离出来,并提供灵活多样的协同方式。

(4) 运行机制自适应 在对外部环境变化的适应能力上,如何保证软件系统在开放、动态、难控环境下具有较好的自适应能力。

(5) 开发方法群体化 从软件开发方式的角度,如何从面向单个程序员的开发方式转变为面向群体最终用户的成长式开发转变。

(6) 外部环境显式化 从外部开放、动态、难控网络环境对软件开发与运行的影响角度,如何对其特征进行分析、建模和处理。

(7) 系统管理自治化 从软件管理的角度,如何确保在开放、动态、难控环境下使得软件系统能够根据需求进行自我管理。

(8) 系统保障可信化 如何在开放、动态、难控环境下为软件系统可信性提供有力可行的支撑。

(9) 核心理论形式化 在面对开放、动态、难控环境下的各种需求,如何为网构软件建立一套合用的、一致的形式化体系,从而形成新一代软件方法学的理论基础。

(10) 技术体系系统化 如何综合上述网构软件不同侧面的方法、技术,面对应用需求,建立系统化的技术体系。

从上述分析来看,运行机制自适应是网构软件核心难点之一,即如何使得网构软件系统能够在运行过程中对外部环境和应用需求的变化作出适当反应,从而使其所提供服务的功能或性能等保持在一个令人满意的水平上^[2]。显然,这一核心问题的解决必须有演化机理和技术的支撑,以实现网构软件的演化,从而适应环境和需求的变化。

软件演化由一系列复杂的变化活动组成,这些变化活动称为演化操作。从演化形态上看网构软件演化有两种,分别为静态演化和动态演化。静态演化主要从非运行时软件版本的更新,软件功能的增加、删除、更新、修改、合并

和分解等方面入手进行研究,使软件进行渐变并达到所希望的形态。动态演化研究软件在运行时刻的变换、更替和升级,主要是指当过程模型发生改变时,通过软件实体运行实例的变换来进行深入研究。

目前,对于软件静态演化研究,主要从体系结构(Software Architecture, SA)的角度进行研究,文献[3]采用可达矩阵对发生变化的构件(演化源)对SA的影响程度(波及效应)进行分析,通过给出构件对SA贡献大小来评价SA演化的波及效应;文献[4]~文献[7]分别用进程代数和图论来解释SA的演化;文献[8]~文献[10]对软件演化过程进行了定性分析。由于网构软件是由来自不同厂商的构件动态组合而成,当某个构件发生演化时,必须针对不同构件协助关系、不同演化操作特征和构件之间不同的关系特性,分析其他构件是否受到影响,并确定影响范围和影响程度,从而明确指出来自哪些厂商的构件需要进行适应性修改,哪些构件不用改变,从而为网构软件静态演化进程的控制奠定基础。

网构软件动态演化研究,从控制流(Control Flow)和数据流(Data Flow)两个方面展开研究^[11]。控制流方面研究致力于保证过程模型实例在动态演化前后控制流的正确性、合理性、一致性^[12~18]。数据流方面的研究主要分为两种方式,一种方式是将变量作为过程模型的一阶实体^[19~21],另一种方式是将活动变量作为标签附加到活动上^[12,15,22~24]。由于网构软件要求实现软件协同的分离化,数据流和控制流的分离是其中一个核心要求,因此,需要从数据流和控制流分离的角度对网构软件的动态演化展开研究。

在我国,自从杨芙清院士首次提出网构软件概念^[25]以来,以北京大学、南京大学为首的多家高校和科研院所,在网构软件实体模型、网构软件实体协同、网构软件运行平台、网构软件开发方法等方面研究取得了重大进展,初步构建了网构软件理论体系框架;同时在网构软件演化方面,从体系结构支持、运行平台支撑和信任机制等方面也取得了很多成果^[26,27]。然而网构软件演化是一个年轻的领域,它关注的焦点以及它的基础概念都在不断变化^[28],在网构软件演化的演化管理、实例迁移和演化准则制约等方面的理论与方法尚不成熟、不完善^[11,26]。因此,迫切需要在网构软件理论体系框架指导下,结合实体主体化、协同分离化等特征,对网构软件静态演化和动态演化技术与方法展开研究,从而有效地完善我国网构软件理论体系,推动我国网构软件向工程化、产业化发展,以提升我国网构软件的国际竞争力。



从网构软件的静态体系结构层面上看,构件之间通过接口产生关系,接口由多个端口组成,接口之间的调用关系传递构件之间的语义,形成构件之间的语义关系,不同接口之间的调用具有不同的时序逻辑性,体现了接口调用的语义协议关系。从运行过程层面上看,网构软件是通过控制流和数据流实现不同构件实体运行实例之间的语义交互关系,从而使网构软件构成一个有机的整体,以运行的角度来看,网构软件的语义关系表现为构件实体运行实例之间的数据流和控制流。

针对上述问题和需求,依据组成网构软件的构件之间语义关系特征,本书将语义、语义协议关系引入到网构软件的静态演化中,通过分析网构软件构件内部与构件之间的语义关系,正确表达构件本身、构件之间的语义关系以及网构软件体系结构的语义关系。然后,提出网构软件中构件的增加、删除、修改等操作对 SA 的影响程度和范围的算法,从而确定网构静态演化的波及效应。在此基础上,通过将数据流显式地引入网构软件建模过程,分析数据流可能导致的动态演化错误,从数据流和控制流的角度,提出网构软件运行实例关于数据流和控制流的动态可迁移性准则,从而确保实现网构软件动态演化实施的合理性、正确性和一致性。

1.2 网构软件演化技术

网构软件能够在开放、动态与难以控制的 Internet 环境下快速地构造应用,并能及时响应内部需求和外部环境的变化,从而具有强大的生命力而被业界接受和采纳。因此,网构软件演化技术对于网构软件发展与应用起着至关重要的作用。网构软件演化的驱动力来自于需求和环境的变更^[2,29],这些驱动力具体表现如下。

(1) 内部需求和环境的变化。包括网构软件的内部执行环境的变化、网构软件组成实体本身的优化和纠错、业务过程的再工程(Business Process Re-engineering)和改变自身经营策略等。

(2) 外部需求和环境的变化。包括变化个性化用户需求、改变伙伴服务的需求和过程协议、变更网构软件的外部执行环境发生、新法律法规的出现、调整商业环境和商业规则等。

这些变化可能发生在网构软件全生命周期的各个阶段,这就要求网构软件应该具有静态(离线)和动态(在线)演化的能力。

在计算机软件领域,软件演化(Software Evolution)是指在软件系统的生命周期内软件维护和软件更新的行为和过程^[30]。对应的网构软件演化技术可以分为静态演化和动态演化。网构软件的静态演化是指根据需求和环境的变化,利用一些演化操作修改其网构软件实体及其过程定义(程序、调用关系等),使得网构软件从一个版本升级到另一个版本的过程。网构软件动态演化是指根据需求和环境的变化,在运行时时刻修改网构软件的过程定义,并将过程定义的变化动态地传播到正在运行的网构软件实体运行实例上的过程。不难发现,网构软件动态演化是以静态演化为基础的,因此在网构软件动态演化的研究中不可避免地涉及网构软件的静态演化。

在开放、动态与难控的 Internet 环境下,网构软件演化必须遵循网构软件的实体主体化、协同分离化和自治化管理等特征和要求。与传统相对封闭、静态、流程固定的经典软件系统相比,实施网构软件的静态演化,首先必须从网构软件体系结构的层面上,精确地指出演化源(即发起变化的构件)对其他构件和整体体系结构是否产生影响,影响程度有多大,要求量化地分析计算 SA 演化的波及效应,由于网构软件中来自其他厂商的构件的源代码不可获取,因此演化波及效应的分析计算只能在构件之间的语义及其协议关系基础上进行;网构软件要求实现数据流和控制流的分离,与传统经典软件系统控制流和数据流大多紧密结合相比,网构软件动态演化,不但必须充分理解控制流、数据流的各自特征,而且需要理解数据流和控制流分离后它们相互作用对动态演化的影响规律,才能有效实施网构软件的动态演化。

1.2.1 网构软件静态演化

现阶段,网构软件初始版本的设计开发一般按以下步骤进行:首先,从用户需求和运行环境入手,采用软件需求分析方法对用户需求进行描述;然后,业务建模人员对需求进行建模并通过与用户的交流对需求进行细化;最后,由软件开发人员将需求映射到过程模型(如 BPEL)层面上,从而形成满足用户需求的网构软件系统。而当需求或环境发生变化时,可以通过两种方式得到新版本:

(1)从最新、最完整的需求和当前的环境下,按照上述思路,获得新的构成



网构软件的软件实体以及相应的软件实体的过程模型。

(2) 仅从不断变化的需求和环境信息着手,将发生变化的要素转化为具体的网构软件演化操作(Evolution Operation),将获得的演化操作作用于先前的网构软件实体及其过程模型,从而得到更新的版本^[31]。

其中,第二种方式真正反映了网构软件演化的特征。网构软件的静态演化需要关注的是网构软件及其实体从一个版本修改或升级为另一个版本的问题,涉及过程模型层面和代码层面。在模型层面,需要关注网构软件实体之间的过程模型变化及其对整体软件系统正确性的影响。而在代码层面,主要是考虑网构软件实体代码修改对整体软件系统正确性的影响。因此静态演化需要考虑以下两个方面的问题:

(1) 在过程模型层面上,网构软件被视为所有参与服务实体(即组成网构软件的所有实体单元)的共享资源,那么各个软件实体对网构软件的修改应该保持互斥。

(2) 在运行时刻,当某一实体的变化(演化源)需要修改全局的网构软件时,它应该确定哪些关联实体(伙伴服务)将会受此影响,哪些不会。

由于网构软件由来自不同厂商的软件实体(统称为构件)组合而成,从软件开发形态和软件体系结构层面上看,具有基于构件的软件工程 CBSE(Component - Based Software Engineering)特征。

1.2.2 网构软件动态演化

在网构软件运行时刻,当需求和环境发生变化时,首先需要修改或升级网构软件的过程模型,然后将模型的变化反映到代码修改中,最后获得新的网构软件版本。当获得新的网构软件版本后,需要将发生在过程模型层面的变化动态地传播到当前正在运行的实例上。具体来说,存在如下三类方式来处理原来的运行实例^[12,17]:

(1) 根据网构软件演化后的模型从头开始执行,这类处理方式丢弃了已执行部分的运行结果。

(2) 根据网构软件演化前的模型继续执行实例,这种模式或者不能使当前的网构软件运行实例体验到新过程模型带来的好处,或者不能满足新的法律法规或新的用户需求。

(3) 尽可能地将当前的网构软件实体运行实例动态地迁移到演化后的过程模型下继续执行,这种模式避免了上述两类处理方式的缺点。

显然网构软件动态演化要求采用第三种方式实现,即需要使用实例迁移机制实现网构软件的动态演化,将正在运行的网构软件运行实例动态地迁移到演化后的过程模型版本上继续执行^[32~34]。而此时,网构软件实体运行实例的迁移需要保证迁移过后不会发生一些动态演化错误^[17,35]。

动态演化是网构软件的一个核心内容,要求在运行时刻、在没有(或尽量少)用户干预的前提下进行演化,从而使其所提供的服务的功能、性能等保持在一个令人满意的水平上^[2,26]。网构软件需要解决的核心难点问题也包括软件实体主体化和软件协同分离化。网构软件要求软件实体具有内容自包含、结构自独立与实体自适应等特性,以便能够适应不断开放的开发与应用环境;软件协同是指软件实体之间通过交互与通信,从而使整个软件系统协同运行,以实现既定目标的过程。软件协同在传统的结构化和面向对象软件技术中处于从属地位,隐藏于计算逻辑之中。而在网构软件形式下,软件实体的计算逻辑所提供的各种服务是高度主体化的,只能在尊重这种主体性的前提下实现协同,这要求软件的协同机制能够从软件实体计算逻辑中分离出来,并且能够适应用户需求和环境的变化^[1,2]。因此,网构软件动态演化必须适应软件实体主体化和软件协同分离化的需求。从高层体系结构上看,软件实体之间的控制语义及其协同过程构成了系统的控制流,软件实体对数据的计算处理逻辑以及实体之间的数据依赖关系构成了系统的数据流,所以软件协同分离化要求必须实现控制流与数据流相互分离。

网构软件大多数需要为用户提供 7×24 的服务。这要求在不中断网构软件所提供服务的条件下,自动或在外部动作指导下实施错误修正、功能完善和性能优化等软件改变活动。

1.3 主要工作与组织结构

当需求和环境产生变化,需要对网构软件实施演化时,首先要确定演化所采取的策略和流程,并对演化策略和流程涉及的演化操作活动及其特性进行详细分析;根据代码修改和演化策略、演化操作活动的性质,对演化所带来的



变化传播特性进行分析,根据分析结果调整需要改变的其他相关构件;然后对需要实施修改的构件进行代码修改并升级构件的代码;如果能够停止当前正在运行的网构软件系统,则停止当前的原软件版本运行,运行新版本。如果网构软件要求提供 7×24 的服务,而不能停止当前的服务,则需要将当前的实例迁移到新的流程模式下运行,实施动态演化,在实施动态演化之前,首先需要分析动态演化是否会产生新的错误(动态演化错误);然后根据可能出现的动态演化错误,制定行之有效的运行实例可迁移性准则,最后实施运行实例的动态迁移。网构软件演化的整体概述如图 1.1 所示。

在上述过程中,针对网构软件演化特性,存在两个核心问题。

(1)静态演化方面 当产生变化的构件(演化源)影响到全局的网构软件整体时,如何确定网构软件中哪些构件受到影响,并界定它对网构软件的体系结构的影响程度和影响范围。

(2)动态演化方面 在遵循网构软件实体主体化、软件协同分离化和核心理论的形式化等特征下,如何将软件实体的运行实例从当前流程模式(源模式)动态地迁移到新流程模式下(目标模式)继续执行,并且满足:①源模式稳态的继承性,即已执行活动的相关数据、状态、关系和结果能够在目标模式下继承,并在后续的活动执行中得到应用;②保证目标模式状态的有效性,不能引入动态演化错误,使目标模式产生死锁、活锁或流程异常终止等现象。

网构软件静态演化研究需要从高层的体系结构出发,而动态演化研究则需要避免动态演化错误的发生。从网构软件体系结构上看,构件之间是通过接口实现交互,从而实现语义的传递。而网构软件动态演化过程中需要关注数据流和控制流的分离。因此本书对于网构软件演化技术研究主要分为以下两个部分。

(1)在对网构软件静态演化的研究中,通过软件语义、语义协议关系和 SA 语义关系模型,研究网构软件体系结构中构件本身及构件之间语义关系及其特征,采用语义协议关系及其强弱特性,分析构件内部和构件之间语义的时序逻辑性,形成对网构软件体系结构的清晰描述,通过构造 SA 的语义关系矩阵和语义关系链矩阵,清晰描述网构软件中某个构件的演化操作引发影响范围,建立语义关系链波及效应指数和语义关系构件波及效应指数,提出网构软件静态演化波及效应的分析计算方法,确定演化操作如构件增加、删除、修改等操作带来的影响程度和范围。