“十一五”国家重点图书　　计算机科学与技术学科前沿丛书
计算机科学与技术学科研究生系列教材（中文版）

面向主体软件工程

——模型、方法学与语言（第2版）

毛新军 编著

清华大学出版社

“十一五”国家重点图书　　计算机科学与技术学科前沿丛书
计算机科学与技术学科研究生系列教材（中文版）

面向主体软件工程

——模型、方法学与语言（第2版）

毛新军 编著

清华大学出版社
北京

内 容 简 介

面向主体软件工程借助于多主体系统的概念、思想、理论和技术来支持软件系统的开发，其基本概念、核心机制、抽象和模型、开发方法学、构造和实现技术等有别于现有的主流软件工程，代表了一种新颖的软件工程范型，可为部署和运行在以互联网和移动互联网为代表的开放环境之上，具有分布、异构、发散、自治、环境敏感、自适应和持续演化等特点的复杂软件系统开发提供有效的技术手段。目前面向主体软件工程已应用于诸如航空、航天、国防、军事、电力、交通、娱乐、游戏、模拟仿真等领域，并为企业计算、面向服务计算、云计算、自适应软件技术、自组织软件工程、信息系统等研究方向提供关键技术支持。本书以多主体系统的具体研究成果为基础，分析了面向主体软件工程的产生和发展背景，阐明了面向主体软件工程的基本思想、哲理和原则，着重从模型、方法学和语言三个方面介绍了面向主体软件工程的具体内容，包括软件体系结构、建模语言、分析和设计方法学、设计模式、程序设计及语言、模型驱动开发、CASE 工具与环境等，结合当前研究和实践状况，讨论了面向主体软件工程存在的问题、面临的挑战以及未来的研究方向。此外，本书还在每一章后面提供了进一步阅读信息，附录部分提供了与面向主体软件工程相关的学术资源信息。

本书可以作为软件工程、人工智能等专业研究生的教材和参考用书，对从事多主体系统、面向主体软件工程等方向研究和实践的人员具有重要的参考价值。

图书在版编目(CIP)数据

面向主体软件工程：模型、方法学与语言/毛新军编著. —2 版. —北京：清华大学出版社，2015
计算机科学与技术学科前沿丛书　大学计算机学科与技术学科研究生系列教材：中文版
ISBN 978-7-302-40341-9

Ⅰ. ①面…　Ⅱ. ①毛…　Ⅲ. ①软件工程—研究生—教材　Ⅳ. ①TP311.5

中国版本图书馆 CIP 数据核字(2015)第 112744 号

责任编辑：张瑞庆　王冰飞
封面设计：傅瑞学
责任校对：焦丽丽
责任印制：何　芊

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　邮　　编：100084
社 总 机：010-62770175　　邮　　购：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质量反馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
课件下载：http://www.tup.com.cn,010-62795954

印 装 者：北京鑫海金澳胶印有限公司
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：26.25　　**字　　数**：641 千字
版　　次：2005 年 6 月第 1 版　2015 年 8 月第 2 版　　**印　　次**：2015 年 8 月第1 次印刷
印　　数：1～2000
定　　价：49.00 元

产品编号：050103-01

Abstract

Agent-Oriented software engineering (AOSE) borrows the concepts, metaphors, theories and technologies of multi-agent system to support the development of software systems. It is essentially different from current mainstream software engineering paradigms in fundamental concepts, abstractions and models, development methodologies, construction and implementation techniques, and represents a new software engineering paradigm that has the potential to provide effective technologies for kinds of complex applications that are deployed in open environment like Internet or mobile Internet, and have the characteristics of distribution, decentralization, autonomy, self-awareness and context-awareness, self-adaptation and continuous evolution. Such software engineering paradigm has been successfully applied in several domains like aerospace, military, electronic power, transportation, amusement and game, simulation, etc. It also provides key technologies and establishes a solid foundation for a number of research branches of computer science, e. g., enterprise computing, service-oriented computing, cloud computing, self-adaptive software, self-organization software engineering, information systems, etc. Based on the theories and technologies of multi-agent system, the book discusses the backgrounds, philosophies and principles of agent-oriented software engineering, details the technical constituents of agent-oriented software engineering from the aspects of models, methodologies and languages, including software architecture, modeling language, analysis and design methodologies, design patterns, programming and languages, model-driven development, CASE tools and environments. The potential problems and challenges, and the future research directions are discussed according to the analysis of the current researches and practices. Moreover, the last section in each chapter provides more information for readers to guide the further readings, the related academic resources are appended. The undergraduates and graduates in software engineering and artificial intelligence can use this book as textbook or reference book. It is also helpful and valuable for the researchers and practitioners involving the researches and practices of multi-agent system and agent-oriented software engineering.

前言

面向主体软件工程是近年来出现的一种新颖软件工程范型，它借助于多主体系统的概念、思想、理论和方法，试图为部署在开放环境，需要与环境进行持续交互，具有异构、分布、发散、自主、自治、协同等特点的软件系统提供工程化的开发手段。随着互联网和移动互联网技术的快速发展以及建立在它们之上应用的急剧增长，越来越多的软件系统呈现出上述特点并且表现为系统之系统、超大规模系统、社会技术系统、动态适应系统等系统形态。面向主体软件工程可以为这类系统的开发提供新颖、有效的技术途径。

面向主体软件工程的研究具有多学科交叉的特点，涉及人工智能、认知科学、自然语言处理、社会组织学等学科的知识；其应用和实践覆盖了诸多领域，包括航空航天、游戏娱乐、国防军事、电子商务、能源电力、医疗健康、机器人和无人设备、电信通信、交通运输、生产物流等，并取得了诸多成功的应用案例，因而受到了学术界和工业界的高度关注和重视，一些政府和组织（如美国、欧盟、NASA、DRAPA、OMG、ISO 等）积极投入该方面的研究，并开展了诸如技术标准化、教育培训、宣传推广等方面的工作。一些国际著名企业也极力推动该方向的研究与应用，包括西门子、戴姆勒、摩托罗拉、IBM、微软、英国电信等。

面向主体软件工程将可能是 21 世纪计算机科学与技术领域的一项重要、富有潜力的技术。随着面向主体软件工程理论和技术的不断发展和日趋成熟，以及在诸多领域的成功实践，它在软件工程领域的地位和作用将日益突出。为了系统了解面向主体软件工程的发展状况、加强面向主体软件工程的教育和培训、推进该方向的科学研究以及在诸多潜在领域的工程实践与应用，我们特编著了本书。

本书内容具有以下几个方面的特点。

（1）系统性。本书从技术和需求两个角度深入分析了面向主体软件工程的产生背景，系统介绍了面向主体软件工程的发展历程，全面阐述了面向主体软件工程的概念和思想、原则和策略、方法学、建模语言、体系结构、设计模式、程序设计、CASE 工具与环境等，并与当前主流软件工程（如面向对象软件工程、面向服务软件工程等）进行了多方面的对比和分析。

（2）前沿性。本书的内容取材于面向主体软件工程近年来的研究成果以及作者在该领域多年研究的心得和体会，展示了该领域的研究状况；书中的许多内容直接来自于我们的科学研究和工程实践成果，反映了作者对面向主体软件工程的认识和看法。此外，我们还对面向主体软件工程研究与实践面临的问题和挑战进行了深入的讨论，指出现有工作存在的局限性和不足，并通过每一章末节的导读部分，帮助读者阅读更多的相关资料和文献，从而引导读者有针对性地开展相关研究和实践。

（3）工程性。除了介绍面向主体软件工程的理论性和方法性内容之外，本书还针对软件工程的实践性特点，强化了对工程要素的介绍，以帮助读者更好地掌握并利用这一技术来

解决实际问题并开展应用开发,包括从工程的角度对比分析了不同技术和方法的优缺点,在各个章节列举了诸多案例来说明如何运用面向主体软件工程的技术和方法,分析了面向主体软件工程的应用情况以及潜在的应用领域,介绍了目前面向主体软件工程应用实践的主要语言、工具和平台,结合软件工程的成功实践经验给出了面向主体软件开发的一组经验和策略。

全书共有11章。第1章从软件工程的角度,结合应用需求以及软件特征的变化,分析面向主体软件工程的产生背景。第2章介绍了主体和多主体系统的概念和思想。第3章概括性地介绍了面向主体软件工程的概念和思想、研究和实践状况。第4章和第5章分别介绍了单个主体的体系结构、多个主体的交互与协同问题及其相应的实现技术。第6章介绍了面向主体的分析和设计方法学。第7章介绍了面向主体的建模语言。第8章介绍了多主体系统的体系结构设计以及面向主体的设计模式。第9章介绍了面向主体的程序设计及其语言。第10章介绍了面向主体的软件开发框架与平台。最后,第11章讨论和分析了面向主体软件工程面临的挑战以及未来的发展方向。本书的附录部分还列举了该领域我们所收集到的一些重要学术资源,包括会议、期刊、学术组织和研究机构。

本书可以作为软件工程专业研究生的教材和参考用书,也可以作为从事多主体系统研究与实践的科研人员、工程开发人员、教师的参考用书。

本书内容得到了国家自然科学基金、国家863计划的赞助和支持,在此表示感谢。由于作者水平有限,书写过程中难免有疏漏之处,敬请读者批评指正。

毛新军

2015年6月于长沙

目录

第1章 绪　论

在计算机科学与技术领域，需求始终是推动计算机科学与技术进步、牵引计算机科学与技术发展的重要驱动力。软件工程发展历程中出现的诸多技术都源自于特定需求的牵引，试图去解决这些需求所带来的问题，并在解决这些问题过程中得到不断的改进、完善和发展。近年来，随着互联网和移动互联网的快速发展和广泛使用，部署在互联网和移动互联网上的软件系统的构成、特点、形态和复杂性发生了深刻的变化。它们以开放的互联网和移动互联网为计算平台和运行环境，通常表现为一类系统之系统(System Of Systems，SOS)、社会技术系统(Socio-Technical System，STS)或者系统联盟(Coalitions Of Systems，COS)，甚至是超大规模系统(Ultra-Large Scale System，ULSS)。系统中的实体具备不同程度的上下文敏感性(Context-Aware)和自我敏感性(Self-Aware)、自主性(Autonomous)、协同性(Cooperative)、自我适应性(Self-Adaptive)和持续演化性(Evolving)等方面的特点，系统的复杂性不仅源自于系统的规模超大，还由于系统及其环境的持续变化和演化以及系统需求的不确定性和动态性。这类系统的出现对软件工程提出了新的需求，也使得现有的软件工程模型、方法、语言和工具等面临着一系列新的挑战，如何支持这类软件系统的分析、设计、构造、部署、运行和演化成为当前软件工程研究与实践迫切需要解决的问题，并将作为一个重要的驱动力促使软件工程的进一步发展及其范型的根本性变化。

本章作为全书的绪论部分，旨在结合需求讨论软件工程的发展及其面临的挑战。本章首先诠释需求变化与技术发展二者之间的内在联系；其次分析近年来软件系统特征和形态发生的深刻变化，阐明这些变化将作为重要驱动力来推动软件工程的发展；第三，介绍了软件工程抽象和技术的发展历程以及近年来出现的一些新颖软件工程范型和技术；第四，以互联网和移动互联网环境下软件系统的变化作为需求，分析这一需求对现有软件工程技术带来的一系列挑战，并依此作为背景诠释面向主体软件工程产生和发展的原因；最后对本章进行小结并推荐进一步阅读的资料。

1.1 需求变化与技术发展

在计算机科学与技术领域，可以发现技术发展与需求变化二者之间存在紧密联系。一方面，需求的出现和变化将牵引和推动技术研究，促进技术的发展和进步，并导致新技术的产生；另一方面，技术的发展和进步将反过来作用于需求，使得各种应用和需求得以拓展和深化，并可能出现新的应用形态，进而促使需求的进

图 1.1　需求变化与技术发展二者之间的关系

一步变化(见图1.1)。例如,早期科学家们在科学研究过程中对信息共享和邮件交换的需求,导致了电子邮件、万维网的产生和发展;而互联网技术的进步使得电子商务、电子政务等应用模式成为可能,并出现了一系列新的应用模式(如网上交易、在线支付、网上商店等)以及由此带来的一系列新的应用需求,如网络安全、基于互联网的身份认证等。类似的典型案例还有很多,例如:

(1) 大规模在线开放课程(Massive Online Open Course,MOOC),在2012年之后的大规模应用是互联网技术和社会计算技术发展到一定阶段的产物,产生了一系列重要的MOOC平台,如Coursera、Udacity和EdX等。MOOC的应用和推广进一步对云计算、服务计算、大数据、群体计算等技术提出了新的需求。

(2) 网上购物,如淘宝、京东网店等,这些应用模式高度依赖于互联网技术、分布计算技术等,并对网络安全、系统集成等技术提出了新的要求。

(3) 微信,近年来微信在移动互联网中获得爆发性的应用,它的出现与移动互联网、智能用户终端、社会计算等技术密切相关,反映了互联网用户在虚拟社会对信息共享和交换的需求。

(4) 软件众包(Software Crowdsourcing),互联网和社会计算技术的发展使得大众通过互联网参与软件众包成为可能,进而促使了一系列应用开发模式(如Apple Store等)以及Topcoder,uTest等软件众包平台的出现。

(5) 互联网技术的发展还催生了一系列新的应用,这些应用的拓展反过来对互联网技术提出了进一步的要求,如近年出现的"快的"软件,它部署在移动互联网上帮助用户方便的打的士等。

在软件工程领域,需求变化与技术发展二者之间的内在关系变得更加明显。为了解决"个体"、"作坊式"的软件开发问题,人们于1968年提出了软件工程的概念和思想,使得软件开发向"工程"和"群体协同工作"方向转化。随后软件工程的研究和实践人员将注意力集中于如何提高软件开发的效率和质量等需求上来。在此需求的驱动下,软件工程研究与实践者在20世纪70年代提出了结构化软件开发方法,并开展了以形式化方法为基础的软件自动化开发等方面的研究。到了20世纪80年代,为了应对日益增长的软件复杂性以及由此带来的软件开发效率低下问题,面向对象软件工程在此背景下应运而生,它提供了高层的抽象、良好的封装和信息隐藏机制、自然建模和继承机制等来加强对复杂软件系统的分析、设计和实现,促进软件重用。到了20世纪90代,以网络技术为基础的分布计算技术发展使得分布式应用成为可能,并在企业计算等领域得到广泛应用,为了解决分布式应用系统的异构性问题以及跨平台的互操作要求,人们提出了COBRA技术规范以及以Java、J2EE为代表的程序设计语言和支撑平台。在这一时期还有另外一项值得关注的软件工程技术——敏捷软件开发方法,它的出现是为了解决以瀑布模型为代表的重型软件开发方法存在的问题,如以软件开发文档为中心、笨重、难以快速应对变化等,而在这一时期人们已经充分认识到软件需求的持续变化在软件开发过程中是客观存在的、必然的和不可避免的,并且软件需求的变化将对软件系统的成功开发带来重要的影响。为了提高软件开发效率和质量,这一时期人们还提出了基于构件软件工程、软件生产线等技术,并加强了诸如软件体系结构和设计模式、模型驱动开发等方面的研究与应用。进入21世纪,为了应对互联网环境下出现的新型软件形态以及由此带来的开发、运行和维护问题,人们提出了一系列新的软件工程技术,如

自治计算的软件工程、面向服务的软件工程、云计算的软件工程、网构软件、自适应软件技术、面向超大规模系统的软件工程、面向主体软件工程、群体化软件工程等。

从本质上看，任何一项有生命力和影响力的理论和技术都有其特定的背景，与特定的需求相联系，并能有效促进与此相关联的一系列问题的解决。在软件工程的发展历程中，软件工程研究与实践者提出了大量的软件工程理论、技术和方法，但是其中的大部分随后消失，留下来的软件工程理论和技术在具体实践和问题解决过程中经受住了检验、解决了问题、发挥了作用，并得到人们的认同和接受。例如，在 20 世纪 80 年代到 90 年代，人们提出了多达几十种面向对象程序设计语言和建模语言，但是目前在学术界和工业界仅仅只有少部分的面向对象程序设计语言和建模语言保留了下来，如 Java、C++ 和 UML，其他的面向对象程序设计语言和建模语言离开了人们的视野并最终成为历史产物，如 Smalltalk 等。因此，软件工程理论和技术的生命力取决于它在多大程度上解决了特定的问题，软件工程研究与实践者在多大程度上接受它。

1.2　软件系统特征的变化

自 20 世纪 90 年代以来，随着互联网技术的快速发展和广泛应用，越来越多的软件系统部署和运行在互联网平台之上。与传统计算环境相比较，互联网环境具有开放、动态、异构等方面的特点，互联网环境中实体的行为以及全局拓扑结构不可预测并且不确定。尤其是进入 21 世纪，移动互联网的快速发展以及各种智能终端和嵌入式设备的广泛使用，运行在互联网和移动互联网之上的软件系统将越来越多的设备、人、社会和政策等紧密的连接和集成在一起，从而使得软件系统中的实体、整个系统的形态和复杂性发生了深刻的变化。

1.2.1　软件环境的变化

回顾计算机的发展历程，可以发现软件的运行和驻留环境大致经历了从主机、个人计算机、局域计算环境、互联网和移动互联网环境的发展历程（见图 1.2）。软件系统的运行环境正经历从集中、封闭和单一的计算环境向发散、开放、动态、多样和异构的互联网环境的转变。

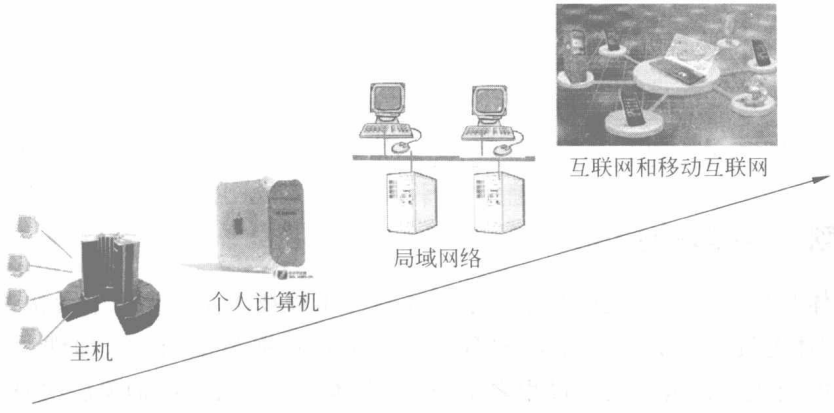

图 1.2　软件系统运行环境的变化

在20世纪90年代之前,大部分软件系统通常被部署在同一个计算设施或者在局域环境中的若干个计算设施之上(如主机或者个人计算机),这些计算设施一般由某个个体或者组织来管理和控制。例如,软件系统被部署在某个主机上,用户通过连接主机的多个终端来访问和操作软件,或者软件被部署在由局域网络连接而成的若干台计算机上(如客户端计算机和服务端计算机),这些软件通过局域网相互交互。它们的部署环境通常是封闭的,不和外界进行交互;环境边界也是明确的,哪些属于环境哪些不属于环境具有清晰的界定。例如,独立的主机或者不联网的个人计算机就构成了封闭的计算环境,局域网计算环境也具有相对封闭性的特点,它无法和局域网之外的计算结点和软件系统进行交互。许多软件系统的部署环境通常是静态、确定和可控的,环境构成要素不会发生变化,或者即使发生变化,这种变化也是可控的。软件系统的行为作用于环境上,对环境产生的影响是确定的。例如,某个软件运行在局域环境中的某个计算结点(如数据库服务器)之上,该服务器要部署哪些软件、什么时候开启和关闭服务器、如何来优化和配置服务器参数等完全由负责该局域环境建设和运维的组织或者机构来决定。如果局域环境发生了某些方面的变化,软件系统的开发和使用人员完全可以掌握、预测和控制这些变化。

进入20世纪90年代,尤其是进入21世纪,越来越多的软件系统被部署在基于互联网的多个不同计算设施之上(甚至包括各种嵌入式设备和移动终端),这些计算设施可能属于不同的组织和机构(如企业、个人、政府等),具有地理上和逻辑上分布、高度自治、独立和发散管理、难以全局控制等方面的特点。互联网是一个动态、难控的环境,不断有新的计算结点、服务和计算资源等加入到互联网中来,或者已有的计算结点、服务和计算资源等会离开互联网环境。此外,互联网环境的动态变化还具有不确定、不可控和不可预测等特点。例如,如果要访问和获取部署在互联网环境的某个服务,但无法预测会有多少个用户和软件同时访问该服务,也无法以一种可控的方式来确定何时能够获得服务以及所获取服务的质量如何。表1.1对比了以个人计算机或者局域计算环境为代表的传统计算环境与互联网环境二者之间的区别。

表1.1 传统计算环境与互联网环境的特点对比

传统计算环境	互联网环境
封闭	开放
静态	动态
确定	不确定
可控	不可控
由特定个人和组织采用集中的方式进行管理	由不同的个人和组织采用发散的方式进行管理

1.2.2 软件实体的变化

软件实体是构成软件系统的基本元素。在软件工程技术及其支撑平台快速发展的驱动之下,软件实体的发展经历了从语句、过程、模块、抽象数据类型、对象、构件、服务等多个不同抽象层次的变化。在这一发展过程之中,软件实体的变化表现出抽象层次越来越高、粒度越来越大、更加便于交互和集成等发展特点。随着大量的软件系统部署在开放、动态和难控的互联网和移动互联网之上,构成这些软件系统的基本软件实体无论在外在特点还是在内

在能力方面都发生了深刻的变化。

1. 异构性

构成软件系统的软件实体通常是异构的，即不同的软件实体可能是由不同的组织和个人，在不同的时间，采用不同的技术（如结构化、面向对象或者服务技术等），借助于不同的语言（如 C、C++ 、Java、Ada 等）、工具（如 Eclipse、Visual Studio）和标准来开发，运行在不同的环境和平台（包括操作系统、虚拟机、中间件和解释器等）之上，并且可能采用不同的数据格式（如文件、数据库等）。对于部署和运行在互联网上的诸多软件系统而言，软件实体的异构性是一种必然。这是因为软件系统的建设、运行、维护和演化通常需要经历很长的一段时间（如几年甚至几十年），在此过程中软件技术在不断的发展、需要持续集成各种遗留的软件系统，因而要采用统一的技术、语言、工具、标准和数据格式等来开发软件几乎是不可能的。

2. 分布性

软件系统拥有大量的软件实体，这些软件实体不仅在形式上是多样的（如表现为不同形式的数据、服务、程序等），而且通常在地理或者逻辑上是分布的，分散部署在互联网环境的不同计算结点之上。例如，企业信息系统所需的、与企业业务密切相关的软件实体被部署在企业内部的业务服务器和数据服务器之上，另外一些与业务相对独立的软件实体（如企业邮箱管理）被托管在其他服务器结点或者云环境上。对于许多互联网软件系统而言，软件实体的分布性是必需的，这是因为越来越多的互联网应用本身就是分布的，此外软件实体的分布性有助于提高软件系统的可靠性和安全性。

3. 发散性

构成软件系统的诸多相互交互的软件实体可能分属于不同的个人、机构和组织，并由他们来进行开发、部署、管理和维护，因而对这些软件实体难以采用传统意义上的集中管理模式，而只能采用发散的方式进行管理。例如，某个企业的电子商务软件需要银行提供的服务来进行交易，需要物流企业提供的服务来实现配送，需要身份认证机构提供的服务进行身份确认。这些负责交易、配送和身份认证的软件实体对于该软件系统而言都是必需的，但是它们不由该电子商务企业来开发、管理和维护。

4. 敏感性

由于软件系统所驻留环境（如互联网）的动态性、开放性和持续演化性，部署其上的软件实体需要具备一定程度的变化敏感性，包括下文敏感性（Context-Aware）和自我敏感性（Self-Aware）。所谓下文敏感性是指软件实体能够感知所在环境的变化，如操作环境、网络环境、应用环境以及与其相关的其他软件实体；自我敏感性是指软件实体能够感知自身状态的变化和行为的实施，如属性取值的变化、软构件的动态加入、软件体系结构的变化、行为的实施等。

5. 协同性

构成软件系统的每个软件实体封装了有限的功能和资源，如有些软件实体提供了用户