

计算机科学与软件研究

——李未院士七十华诞纪念文集

怀进鹏 徐宝文 许可 主编



科学出版社

计算机科学与软件研究

——李未院士七十华诞纪念文集

怀进鹏 徐宝文 许 可 主编



科学出版社

北 京

内 容 简 介

为庆祝李未院士七十华诞,本书收录国内外学者撰写的计算机科学与软件研究领域的论文 49 篇,主要反映这些学者近年来在计算机基础与理论、计算机系统与结构、软件技术、网络与大数据、计算机应用技术等方面所取得的研究成果,以及对相关领域的综述与总结。

本书可供从事计算机科学和软件研究的科研人员、工程技术人员参考,也可供相关学科的研究人员、大专院校师生阅读。

图书在版编目(CIP)数据

计算机科学与软件研究:李未院士七十华诞纪念文集/怀进鹏,徐宝文,许可主编. —北京:科学出版社,2013

ISBN 978-7-03-037473-8

I. ①计… II. ①怀… ②徐… ③许… III. ①计算机科学—文集
②软件工程—文集 IV. ①TP3

中国版本图书馆 CIP 数据核字(2013)第 098307 号

责任编辑:张 濮 任 静 陈 静/责任校对:钟 洋 刘小梅

责任印制:张 倩/封面设计:张 放

科学出版社出版

北京东黄城根北街 16 号

邮政编码:100717

<http://www.sciencep.com>

印刷

科学出版社发行 各地新华书店经销

*

2013 年 6 月第 一 版 开本:16 (787×1092)

2013 年 6 月第一次印刷 印张:50 1/2

字数:1 280 000

定价:268.00 元

(如有印装质量问题,我社负责调换)

问学求道五则*

(代序)

李 未

今天，我讲一讲下述“问学求道”的五个原则，与各位老师和同学共勉。

一、思想境界——忠恕为上

《论语》中有两句话，一句是“己欲立而立人，己欲达而达人”，另一句是“己所不欲，勿施于人”。前一句是说：自己想要立足的，也允许别人立足；如果自己希望自由地发展，那么也要允许别人自由地发展。第二句是说：自己不喜欢、不能忍受的，就不要强迫别人喜欢和忍受。

试想，如果我们周围的每一个人都这么做，那么我们所在的社区和人群就是和谐社区和人群。如果中国所有的社区和人群都是和谐的，那么我们就生活在和谐社会之中。如果全世界的人们都能够这么做，人类就进入大同世界了。

马克思、恩格斯也持有相同的思想。他们在《共产党宣言》里有一段对理想社会的描述：“代替那纯粹的阶级和阶级对立的资产阶级旧社会的，是这样的一个联合体，在这个联合体中，每一个人的自由发展是他人发展的条件。”这说明人类在这一根本点上有共同的认识和追求。我们每个人只要从自己做起，社会就会变得更加美好。

二、人生态度——勤奋乐观

孟子说过：“天将降大任于斯人也”，所以“必先苦其心智，劳其筋骨，饿其体肤，空乏其身，行拂乱其所为，所以动心忍性，增益其所不能”。要担当推动人类进步的责任，就要严格地要求自己，就要付出艰苦的努力。但是，也要保持一个好的心情、乐观的生活态度。歌德说：“辽阔的世界，宏伟的人生，长年累月，真诚地勤奋，不断地探索，不断地创新，常常周而复始，但从不停顿，尊重传统，而又乐于创新，心情舒畅，目标纯正，啊！这样又会前进一程！”如果我们能把东方和西方两位先哲对生活的态度结合起来，并身体力行，那么我们就不会浪费生命中任何一点时间，乐观而有意义地生活。

三、治学之道——一万小时

我常想给出一个做学问的量化标准，曾提出过“三倍于常人的努力”来鼓励自己和学生。最近读了《异类》这本书，很受启发。作者格拉德威尔提出了“一万小时”的概念。我

* 本文曾以同名刊载于《变革中的大学——李未教育文存》（高等教育出版社，2012）。

为庆贺李未院士70寿辰，2012年9月8日，北京航空航天大学计算机学院与软件开发环境国家重点实验室联合主办了“计算机科学与软件工程发展论坛”。国内外40余所高校、科研院所的70多位知名中青年学者、北京航空航天大学数十位教师和200余名研究生欢聚一堂，共同探讨计算机科学和软件工程的发展问题，其中32位学者作了发言。看到那么多中青年才俊济济一堂，李未院士感到由衷的高兴，首次向大家讲述自己在问学求道之路上的五点体会，以与大家共勉。

本论文集是在会后向部分与会学者征集文章而成的，共计收入49篇论文，分为综述、基础与理论、系统与结构、软件技术、网络与大数据、应用技术等六个栏目。编者在征得李未院士同意后将其五点体会作为本论文集的序。

理解就是在人生的任何一个阶段，不论自己做什么工作，都要选准一个特定的问题。这个问题一定是你最感兴趣的问题，和当前的工作一致更好，不一致也没有关系。对这个问题，要用不少于一万小时的时间进行研究。在这一万小时的研究中，使自己对这个问题的认识，从生疏到熟悉，从熟到巧、到精，一直升华到一个境界。在这个境界中，能看到别人所看不见、所认识不到的东西。在这个时候，自己说出来的话、提出的理论、发明的技术、做出的研究成果，就是创新。一万小时并不长，一个博士生，如果每天在这个特定的问题上花 6 个小时进行研究，5 年就是一万小时。一万小时之后，一旦在这个问题上取得了创新成果，整个人对事物的认识就会升华到一个新阶段，是一个在哲学和方法论层面对一般规律认识的阶段。那个时候无论做什么事情，都能够触类旁通，从整体上提升处理人生各种问题的高度和水平。

四、交流沟通——与人为善

面对交流的对象，不论群体，还是个人，要站在对方的立场，用自己的眼光，提出建设性的意见。

站在对方的立场，要求自己首先理解对方。这样，到自己发表意见时，才能让对方能够理解和接受。运用自己的眼光，就是从一个新的视角来分析和研究问题，这样就可以看到对方没有想到和看到的新意。这种新意可以用两种态度表达出来。一种是批判的态度，指出其不足。另外一种是建设性的态度，在给出改进的方法和建设性意见之中，消除不足于无形。建设性的态度使意见容易被接受，即使不被接受，也不会产生新的矛盾。总之，站在对方的立场用自己的眼光发表建设性的意见，有百利而无一害。

五、工作态度——闻过则喜

孔子讲过，人有生而知之者，也有学而知之者。他说自己是学而知之者。其实，我们绝大多数人都是学而知之者。我们对于一个事物都有一个学习和认识的过程，而且我们对这个问题的认识常常要反复多次。所谓反复就是出现了挫折或犯了错误。犯了错误不要紧，及时改正是最要紧的。改正了就是前进了，改正了就可以避免不再重犯这个错误，改正了就是创新、就是创造。孟子讲“闻过则喜”，现在讲“变坏事为好事”，都是这个道理。因此，从这个角度看，遇到挫折是好事，犯错误没有关系，要知错即改正，我们就会有创新，我们就前进了。

以上五条就是我在自己问学求道的过程中总结出来的几条原则。我即将七十岁，这五条自认为仍没有做好，我要继续努力，活到老学到老。

目 录

问学求道五则（代序）

综 述

面向网络的软件技术：现状与趋势	梅 宏, 刘譞哲	3
人工智能发展趋势：混合智能研究	吴朝晖, 潘 纲, 郑能干	21
SMT 及其扩展问题	马菲菲, 张 健	29
Service-Oriented Development	WANG Andy, ZHENG Guangzhi	36
Intelligent Transportation Systems for Smart Cities: A Progress Review		
.....XIONG Zhang, SHENG Hao, RONG Wenge, COOPER Dave E		52
浅谈跨学科计算思维教学的认识与实践	李晓明	61

基础与理论

New Algorithmic Techniques in Parameterized Computation	CHEN Jian'er	67
Simultaneous Maximal Contractions and Simultaneously Maximally Consistent Subsets of Theories	SUI Yuefei, LI Wei	76
Local Search with Edge Weighting and Configuration Checking Heuristics for Minimum Vertex Cover	CAI Shaowei, SU Kaile, SATTAR Abdul	86
Automata-Based Analysis of Stage Suspended Boom Systems		
.....HE Anping, WU Jinzhao, ZHOU Yongquan, YANG Shihan		122
一类异步连接的消息序列图规约的时间性质分析	潘敏学, 李宣东	134
中介真值程度的度量及其应用的回顾	洪 龙	146
On the Decidability of Open Logic	ZHANG Wei	156
From Quantum Computing Device to Quantum Software		
.....WU Nan, HU Haixing, SONG Fangmin, ZHENG Huimin, LI Xiangdong		167
Formal Semantics and Verification of AADL Modes in Timed Abstract State Machine		
.....YANG Zhibin, HU Kai, MA Dianfu, PI Lei, BODEVELX Jean-Paul		178
安全苛刻系统自动化测试的形式化语义模型及性质	吕江花, 马世龙, 李先军	190
R-calculus without the Cut Rule	LUO Jie, LI Wei	207

系统与结构

桌面虚拟化: 探索与实现	廖小飞, 金海, 刘海坤, 邓莉	225
Using Kernel Coupling to Predict the Performance of Parallel Applications	WU Xingfu, TAYLOR Valerie, GEISLER Jonathan, STEVENS Rick	238
一种基于光谱空间几何特性的高光谱图像快速鲁棒解混方法	张艳宁, 张海超, 段锋	259
Adaptive Controller for Dynamic Power and Performance Management in the Virtualized Computing Systems	WEN Chengjian, LONG Xiang, MU Yifen	269
A Flexible Attribute Based Access Control Method for Grid Computing	LANG Bo, FOSTER Ian, SIEBENLIST Frank, ANANTHAKRISHNAN Rachana, FREEMAN Tim	284
Simulation of Power Consumption of Cloud Data Centers	LUO Liang, WU Wenjun, TSAI Wei-Tek, DI Dichen, ZHANG Fei	298

软件技术

A Tetrahedral Data Model for Unstructured Data Management	LI Wei, LANG Bo	327
Internetwork: A Shift of Software Paradigm	LV Jian, MA Xiaoxing, HUANG Yu, CAO Chun, XU Feng	343
StarBus+: Distributed Object Middleware Practice for Internet Computing	WANG Huaimin, WANG Yufeng, TANG Yangbin	359
XML 语义定义语言	林作铨, 刘升平, 李未	374
Crosstalk Aware Register Reallocation Method for Green Compiling	CHEN Yong, HE Yanxiang, XU Chao, CHEN Nian, WU Wei, LI Qing'an	397
Automated Program Debugging via Multiple Predicate Switching	LIU Yongmei, LI Bing	409
使用 PTCCS 的 Web 服务组合功能性和非功能性方面的统一建模	黄志球, 肖芳雄	420
面向特征的软件分析与测试方法研究	刘超, 晏海华, 何智涛	446
Examining the Potentially Confounding Effect of Class Size on the Associations between Object-Oriented Metrics and Change-Proneness	ZHOU Yuming, LEUNG Hareton, XU Baowen	462
A Revisit of Fault Class Hierarchies in General Boolean Specifications	CHEN Zhenyu, CHEN Tsongyueh, XU Baowen	489

网络与大数据

(ϵ, δ)-Approximate Aggregation Algorithms in Dynamic Sensor Networks	503
..... <i>LI Jianzhong, CHENG Siyao</i>	
Graph Pattern Matching Revised for Social Network Analysis	523
..... <i>FAN Wenfei</i>	
The Small-Community Phenomenon in Networks	548
..... <i>LI Angsheng, PENG Pan</i>	
制造网格中的自动双赢服务协商模型	578
..... 孟祥旭, 潘 丽	
基于模型检测的无线传感网安全协议形式化分析与改进	587
..... 陈铁明, 何卡特, 江 颀	
Block Compressive Sensing in Data Gathering for Large-Scale Wireless Sensor Networks	
..... <i>LIU Yanlei, LI Huan</i>	597
Distributed Graph Pattern Matching	611
..... <i>MA Shuai, CAO Yang, HUAI Jinpeng, WO Tianyu</i>	
Inconsistency between Maximum Information Entropy and Homophily in Social Networks	630
..... <i>ZHAO Jichang, LIANG Xiao, XU Ke</i>	

应 用 技 术

Region-Based Rate-Control for H.264/AVC for Low Bit-Rate Applications	
..... <i>HU Haimiao, LI Bo, LIN Weiyao, LI Wei, SUN Mingting</i>	639
A Heuristic Path-Estimating Algorithm for Large-Scale Real-Time Traffic Information Calculating	662
..... <i>LV Weifeng, ZHU Tongyu, WU Dongdong, DAI Hong, HUANG Jian</i>	
Combining Tensor Space Analysis and Active Appearance Models for Aging Effect Simulation on Face Images	
..... <i>WANG Yunhong, ZHANG Zhaoxiang, LI Weixin, JIANG Fangyuan</i>	672
TL-Tree: Flash-Optimized Storage for Time-Series Sensing Data on Sensor Platforms	
..... <i>LI Huan, LIANG Dong, XIE Lihui, ZHANG Gong, RAMAMRITHAM Krithi</i>	693
Three-Dimensional Deformable-Model-Based Localization and Recognition of Road Vehicles	708
..... <i>ZHANG Zhaoxiang, TAN Tieniu, HUANG Kaiqi, WANG Yunhong</i>	
Robust Multiple-Vehicle Tracking via Adaptive Integration of Multiple Visual Features	
..... <i>SHENG Hao, WEI Qi, LI Chao, XIONG Zhang</i>	731
Model Order Selection in Reversible Image Watermarking	
..... <i>CHEN Ming, CHEN Zhenyong, ZENG Xiao, XIONG Zhang</i>	753
3-D Face Recognition Using eLBP-Based Facial Description and Local Feature Hybrid Matching	775
..... <i>HUANG Di, ARDABILIAN Mohsen, WANG Yunhong, CHEN Liming</i>	

综 述



面向网络的软件技术：现状与趋势*

梅 宏^{1,2} 刘譞哲^{1,2}

(1. 北京大学信息科学技术学院软件研究所;
2. 高可信软件技术教育部重点实验室)

摘 要 目前，以互联网为主干，电信网、广电网、传感网等多种网络正在不断交叉渗透，信息世界、人类社会、物理世界之间趋于融合，为信息技术及其应用开辟了更为广阔的发展空间。这对传统软件理论、方法和技术带来了一系列的挑战。本文回顾了现有基于网络的新型应用模式，探讨了软件模型、软件运行支撑机制与平台、软件开发方法和软件质量评价与保障体系的现状和发展趋势，并介绍了国内的研究现状。

关键词 网络化软件；软件范型；软件运行支撑机制与平台；软件开发方法；软件质量度量与评估

1 引言

计算机软件技术主要涉及四个方面的内容（见图 1）：软件范型、软件开发（构造）方法、软件运行支撑及软件质量度量与评估。其中，软件范型是从软件工程师（或程序员）视角看到的软件模型及其构造原理，是软件技术体系的核心。回顾软件技术发展历程，软件范型大致经历了无结构、结构化、面向对象、面向构件/面向服务等变迁，而每次软件范型的变迁，都会引发软件开发方法和运行支撑机制的相应变化，并导致新的软件质量度量和评估方法的出现。

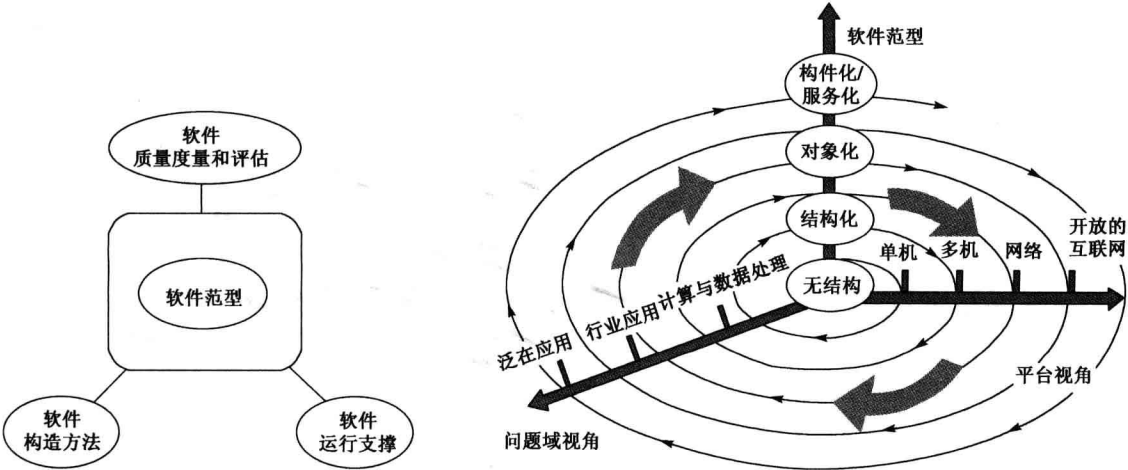


图 1 软件技术体系

图 2 软件范型的变迁

本质上，软件范型及其支撑技术体系的演变，反映了软件技术发展的基本驱动力，即，

* 本文早期版本发表于《科学通报》2010 年第 13 期。本文以该文为基础，结合软件技术领域近两年来的新进展，进行了修订和补充。

追求更具表达能力、更符合人类思维模式、易构造、易演化的软件模型；支持高效率和高质量的软件开发；充分发挥硬件资源的能力，支持高效能、高可靠和易管理的软件运行。

在计算技术发展的较长时期，软件技术研究主要针对单机系统，软件范型从无结构演变到结构化和面向对象，为软件解空间和问题空间的复杂性控制提供了有效机制和手段；软件开发技术则主要围绕相应的软件范型，研究大型软件系统高质量高效率开发的方法、技术和工具；在运行支撑方面，面向单机的操作系统主要随着计算机硬件的发展而进步，追求更高效地发挥底层硬件资源所提供的计算能力，不断凝练应用共性并沉淀为操作系统的固有成分，同时更好地满足用户对易用性的需求；软件质量度量 and 评估方面则主要关注系统质量。

近年来，以互联网为主干，电信网、广电网、传感网等多种网络正在不断交叉渗透，信息世界、人类社会、物理世界之间趋于融合，为信息技术及其应用开辟了更为广阔的发展空间。相应地，计算机软件的运行环境也开始从单机和局域网环境，延展到复杂网络环境；应用领域也从支撑各个领域的行业应用，发展到支持人-机-物融合的网络化、智能化、普适化的应用。互联网平台自身开放、动态、难控的特性，以及新型应用模式和计算模式（如云计算、服务计算、普适计算等）的不断涌现，使得软件形态发生了很大变化；逐渐呈现出自主性、协同性、演化性、情境性、自发性等新的特征，并对可信性提出了很高要求。

在人-机-物融合的网络环境下，针对软件网构化、服务化、泛在化、高可信的发展趋势，亟需新的软件范型，也对软件开发方法、运行支撑平台、质量度量评估及保障机制提出了一系列新的挑战。在软件开发方法方面，随着互联网上可复用软件资源（如软件服务、开源软件等）的不断丰富，基于构件/服务组装的软件开发成为主要途径，出现了全球化软件开发等新型软件开发模式，数据分析、处理与应用在软件开发中的作用日益重要。软件运行平台向着支持网络化和网络应用发展，除了在传统单机操作系统之上出现了软件中间件等新的系统软件层之外，新型网络化操作系统也引起广泛关注，呈现出“云-端”融合的发展趋势，需要管理更大规模的资源（包括计算资源和存储资源）、支持更加多样化的应用、支持新型的应用模式和计算模式、支持用户以多种不同终端设备访问所需的服务。软件质量的关注重点向使用质量转移，可信、绿色等一些新的质量属性开始受到重视。

本文将探讨网络环境下软件技术的现状和发展。首先，回顾了现有基于网络的主要应用模式和计算模式。其次，围绕软件范型中四个方面——软件形态与模型、软件开发方法、软件运行平台和软件质量度量与保障，分析了相关理论、方法和技术的现状以及发展趋势。最后，简要介绍了国内的研究现状。

2 新型网络应用模式和计算模式

互联网产生至今，已经出现了多种基于网络的新型应用模式和计算模式。下面简要回顾若干热点概念及其相关技术的进展情况。

2.1 云计算

云计算是当前网络环境下最为重要的应用模式。云计算的理念与 20 世纪 90 年代中期提出的网格计算（Grid Computing）有很多相似之处^[1]，都是力图整合各类资源并为用户提供一体化透明的服务。网格计算的目标，是整合分布于互联网上的计算资源、存储资源、通信资源、软件资源和知识资源等。而相对于网格计算中资源地域分布的特点，云计算则基于数据中心（Data Center），

以集中式的方式汇聚各类计算资源、存储资源和软件资源,以服务的形式供用户在线使用。从这个意义上看,软件服务化是云计算的核心使能技术之一^[2]。从类型上划分,目前云计算中的服务主要包括“基础设施即服务”(Infrastructure as a Service, IaaS)、“平台即服务”(Platform as a Service, PaaS)和“软件即服务”(Software as a Service)。随着互联网的迅速普及,诸多国际 IT 巨头都开始将云计算作为其产业布局中最为重要的一环,结合自身特点推出了各自的云计算业务。其中,在 IaaS 方面,Amazon 的云计算平台主要提供计算(Elastic Compute Cloud, EC2)和存储服务(Simple Storage Service, S3);IBM 推出的“蓝云”计算平台,提供一系列的自动化、自我管理和自我修复的虚拟化技术。在 PaaS 方面,谷歌推出的 Google AppEngine 平台和百度推出的百度应用引擎(Baidu AppEngine),为开发者提供了在线的开发环境以及编译、调试、部署和运维服务。在 SaaS 方面,微软的 Software + Service 模式、谷歌提供的办公类和生活类系列在线服务、Salesforce.com 提供的在线客户关系管理(CRM)服务等,都已被广泛应用。

云计算环境下,对各类资源进行有效管理是最为核心的挑战,而软件技术则是资源管理的基础支撑。目前,云资源的管理主要包括存储资源、计算资源以及网络资源三个方面。

(1) 在数据存储与管理方面,云计算依托数据中心,通常要存储和维护数以 PB(千万亿字节)计的数据,体量规模更大,结构更加复杂(往往是非结构或半结构化数据),对处理速度的要求也更高,对数据的存取机制、效率、性能、安全和可靠性等方面都提出了新的挑战。因此,出现了一些新型网络化文件管理系统(如 Google File System)^[3]、数据存储和管理机制(如 Big Table^[3]和 Amazon S3^[4]等)、数据分析处理和性能优化技术(如 Map/Reduce^[5]、Hadoop^[6])等。特别地,大数据(Big Data)^[7-8]近期受到了广泛关注,支持大数据的存储、分析和处理,被认为是云计算环境下数据管理与处理未来重要的发展方向。

(2) 在计算资源管理方面,为了提高资源的管理效率,虚拟化技术得到了广泛应用。虚拟化技术支持计算资源的统一管理、负载均衡、可伸缩调度、在线迁移等,是云计算的另一核心使能技术。目前,虚拟化技术的研究主要关注虚拟机的效率、安全和节能等方面,如支持多租户的嵌套虚拟化技术(Nested Virtualization)、虚拟机监控器中的 I/O 资源分配技术和优化算法、虚拟机的安全监控技术等^{[9][10]}。

(3) 在网络管理方面,针对数据中心上网络受到硬件制约和相对静态的问题,近年来软件定义的网络(Software-Defined Networks, SDN)技术开始引起关注,其代表如 OpenFlow^[11]。SDN 通过在交换机和路由器硬件之上搭建软件层,实现了数据转发和路由控制的分离,提高了网络资源的使用效率和数据传输性能,并允许云服务提供者自定义网络功能。

此外,还需要注意的是,尽管目前云资源管理主要围绕公有云开展,但一些服务提供商也开始关注将这些技术用于私有云的资源管理,针对企业计算的实际需求来整合和管理云资源,如开源项目 OpenStack^[12]和 Eucalyptus 公司均在私有云构建上进行了尝试。

2.2 服务计算

服务计算(Service Computing)的概念在 20 世纪 90 年代末期提出,其目标是以服务作为应用构造的基本单元,能够快速、便捷地开发和组装应用系统,以解决在分布、异构的环境中数据、应用和系统集成的问题。

作为服务计算中最为核心理念,软件服务被认为是互联网环境下最为重要的一种形态。软件服务是指将软件的功能以服务的形式通过互联网来交付。软件由服务的运营商负责运维,用户无须拥有软件的本地拷贝,也不必考虑软件的维护和升级,还可以按需定制和购买软件。

这种“只求使用、不求拥有”的软件使用模式，只有软件和互联网的结合才可能实现。目前，软件服务被认为是互联网环境下软件最为主要的形态，也使得软件产业开始从“以产品为中心的制造业”向“以用户为中心的服务业”转变。

软件服务一般基于可共享和集成的资源构建，对外则表现为一组相对独立的业务功能单元（通常是可供外部直接调用的接口），更加方便使用者直接使用。近十年来，在软件服务研究方面，面向服务的体系结构及其主要的实现技术——Web 服务，一度成为学术界和工业界的研究热点。围绕 Web 服务互操作性、发现、组装、语义、管理和质量保障等问题，已经有了大量的工作。但是，从实际应用情况看，目前互联网上真正可用的 Web 服务数量还比较少，且大多只提供相对简单的数据查询功能（如天气、股票查询等），Web 服务之间也很少构成复杂的业务逻辑。这里既有 Web 服务技术体系存在自身缺陷的原因，如广域网中基于 SOAP 协议的消息传输性能较差、服务质量难以保障等，也有一些非技术的因素，如 Web 服务相关的技术标准过于繁杂、面向互联网的服务注册中心（UDDI）难于管理和维护等。

事实上，近年来，很多软件服务提供商（主要是互联网公司，如谷歌、微软、eBay、亚马逊、Facebook、Salesforce.com、百度、新浪等），都开始采用遵循 REST (Representational State Transfer) 体系结构风格的 Web API^①来发布软件服务供用户在线使用^[13]。和基于 SOAP 的 Web 服务相比，基于 REST 的 Web 服务提供了一种相对轻量的实现方式，通过描述资源状态变化的方式而不是远程过程调用（Remote Procedure Call）的方式来使用软件服务，更加适合广域网环境。此外，基于 REST 的 Web 服务功能也不仅限于简单的数据查询，应用的业务范围更广，如谷歌的 Gmail 服务、Salesforce.com 的 CRM（客户关系管理）服务和亚马逊的存储服务，都是基于 REST 的 Web 服务。

近年来，云计算开始成为互联网环境下主要的应用模式，而软件服务则是云计算的基础使能技术之一^[1]。在云计算环境下，软件服务仍有很多问题需要解决，包括对遗产系统“解耦”和服务化封装（特别是发布业务服务而不只是单纯的数据服务）、服务的交付与展现、服务动态组装、服务的跨域安全性、服务资源管理等。

2.3 普适计算

普适计算（Pervasive Computing，相似概念有 Ubiquitous Computing 和 Ambient Computing）的概念在 20 世纪 90 年代初期提出，其目标是无所不在的、随时随地可以进行计算的一种方式。它试图把信息空间与人们生活的物理空间集成为一个整体，从而使人们能够以“5A”方式（即，任何人（Anybody）在任何地点（Anywhere）、任何时间（Anytime）可以使用任何设备（Any device）访问任何信息资源（Anything）使用计算设备。因此，普适计算可视为是从人机交互的角度看待网络系统及其应用的一种计算模式。

近年来，随着人-机-物融合的新型网络环境不断发展，在普适计算研究方面，信息物理融合系统（Cyber Physical System, CPS）得到了广泛的关注。CPS 将信息系统和物理系统集成在一起，形成一种综合计算、网络和物理环境的多时空复杂系统。在 CPS 中，离散的计算过程和连续的物理过程深度融合，计算通过反馈环路控制物理过程，同时，物理环境的变化也在影响着计算。同传统的嵌入式系统主要强调计算能力相比，CPS 则将计算、通信和控制结合在一起，形成了泛在的智能化物联网。CPS 被认为是后工业时代的一项重要技术革命，主要发达国家均开始将 CPS 作为未来具有重大战略意义的研究内容。目前，CPS 的研究尚处

① 通常，基于 REST 体系结构风格的 Web 服务也被称为 RESTful Web 服务。

在起步阶段,主要研究关注点包括支持物理过程与计算过程深度融合的 CPS 系统建模方法、CPS 系统关键属性建模与分析、软件使能的控制与通信方法与技术、基于模型的 CPS 系统动态行为仿真验证方法、CPS 系统实体仿真验证框架与技术等。

2.4 网构软件

上述产生于互联网环境下的新型应用模式和计算模式均是从某种视角或层次对基于网络的应用提出新的理念和技术体系。例如,云计算和网格计算都关注从资源共享与管理的角度探讨未来网络系统的应用与构造模式;普适计算主要关注从人机交互的角度来探讨未来网络系统的应用模式。本质上,这些应用模式和计算模式均离不开软件技术作为其基础支撑,这就需要软件的基本形态和特征、概念框架、逻辑内涵、质量目标等都发生相应的变化,同时也对软件方法与技术提出了全方位的挑战。

我国学者从软件技术角度出发,于 2002 年正式提出了网构软件(Internetware)^[14]的概念。网构软件是互联网环境下的一种新型软件范型^[15]。一方面,网构软件是传统软件在互联网环境下的自然延伸;另一方面,为了适应开放、动态、多变的互联网环境及应用领域全球化、个性化、持续成长等特点,网构软件也有其独有的基本特征,如自主性(软件实体具有相对独立性、主动性和自适应性)、协同性(软件实体之间多种方式的互连、互通、协作和联盟)、演化性(软件实体的个数可变、结构关系和形态动态可调)、情境性(软件实体具有感知外部运行和使用环境的能力)、自发性(软件实体之间的交互往往是非预设、不确定、按需应变的,可能隐含非预期的行为和结果)等。经过十余年的努力,网构软件的研究与实践已取得了一些进展,形成了一套以软件体系结构为中心的技术体系^[15-16],主要包括:在网构软件模型方面,通过开放协同模型、情境驱动模型和智能可信模型,实现了网构软件基本实体的构件化、主体化、服务化以及实体间开放的结构化的按需协同;在网构软件运行平台方面,以构件化的中间件平台体系为基础,实现了支持网构软件基本实体运行的容器、支持按需协同的运行时代体系结构,以及基于反射的网构软件自治管理;在网构软件开发方法方面,以全生命周期软件体系结构为核心,通过模型驱动的方式支持网构软件基本实体和结构化协同的高效开发,以及遗产系统向网构软件的半自动转换,并通过领域建模实现了网构软件无序资源的有序组织。

3 软件模型和编程语言

3.1 软件模型

计算机软件本质上可以看成是现实世界中的问题及其求解方案在计算机上的符号表示。如何将现实世界表示为人和计算机都能够理解的符号系统是计算机软件领域最核心的问题之一,此即软件模型及其建模问题。一方面,软件模型是现实世界的抽象表示,另一方面,它又是现实世界问题及其求解方案在计算机上实现(即计算机软件)的抽象规约。换言之,软件模型是将现实世界问题空间映射到计算机世界解空间的“桥梁”。考察一个软件模型主要是针对两个基本性质:可表达性和可理解性。所谓可表达性,是指软件模型能否准确、全面地表示问题空间。通常,由于问题空间的固有复杂性,很难对问题空间进行准确、全面的建模,只能是一个“尽可能好”的抽象。所谓可理解性,一方面指该模型便于人理解,另一方面也

指该模型便于被计算机系统理解（也即该模型的软件实现能够被计算机系统执行）。由于执行软件的计算机系统能力的有限性，反过来也就影响了模型的表达能力。一个好的软件模型需要在这两个基本性质间的平衡。模型过于复杂，既难于人理解，也难于机器实现和执行；模型易于实现和执行，其表达能力必然有限。

软件模型的发展主要是源自高级程序设计语言的发展和软件范型的发展。回顾过去，软件模型始终在追求更具表达能力、更符合人类思维模式、易构造、易演化的目标。形式上，软件模型是对软件实体及其交互关系的描述^[17]。所谓软件实体，是指构成软件系统的基本元素。过去几十年来，主流软件实体的发展经历了指令、语句、函数/过程、模块、抽象数据类型/对象、构件、服务等粒度和层次，实体间的交互方式也在相应地发生改变。基于硬件“裸机”进行编程，开发人员使用的是机器语言和汇编语言，面对的软件实体是机器指令，实体间交互是指令的顺序执行和转移。基于高级语言虚拟机进行软件开发时，开发人员面对的是软件实体是高级语言语句，实体间的交互表现为顺序结构、分支结构和循环结构三种控制结构。结构化软件范型是基于过程抽象，体现为面向功能的软件模型，软件实体表现为功能模块（函数或过程集合），实体之间通过函数或过程调用集成在一起形成软件系统，其重点在于反映现实世界的功能性。面向对象范型是基于数据抽象，软件实体表现为对象，对象之间通过关联（如继承、集聚等）和消息传递相互联系在一起，其重点在于反映现实世界的结构性和功能性。基于构件的范型是面向对象范型的延伸，软件实体表现为构件，强调功能的相对独立性、自描述性和可复用性，构件通过连接器互联。而面向服务的软件范型则体现了网络时代软件使用方式的重要变化，不再强调对软件代码的拥有，只关注获得软件提供的能力，软件实体通过标准接口对外提供服务，服务请求者（人或软件实体）通过发送服务请求获得所需的计算结果。

近年来，随着互联网及其上应用的快速发展与普及，计算机软件所面临的环境开始从静态封闭逐步走向开放、动态和多变^[18]。网络环境下的软件系统正在呈现出很多新的特征，如自主性、协同性、演化性、情境性、自发性等，这将导致软件系统既可能没有明确的边界，也可能没有稳定的结构。

网络环境下软件的新特征对软件模型提出了新的挑战。一方面，软件模型需要具有更强的表达能力：既能表达软件实体的局部自主性，又能表达由不确定、不可控、真分布的软件实体构成的软件系统的全局稳定性、全局目标可控性，以及目标实现过程的协同性。另一方面，计算系统不再局限于一台计算机，而是开放的网络环境可能被视为一台全局性的虚拟计算机，软件模型不仅可被一台计算机所理解，还需被开放环境下的虚拟计算系统所理解。软件系统的性质越来越具有动态性，甚至其功能性和结构性也不再是一成不变。在一定意义下，软件模型及其实现仅是对现实世界的静态表示，如何用静态的表示手段来刻画现实问题空间的动态性质，是颇具挑战性的难题。

从目前软件模型相关的研究与实践来看，为了适应互联网环境下软件系统的新特征，在软件实体的功能内聚度不断提高、自主性不断增强的基础上，有两个趋势值得关注^[19]：

(1) 随着软件实体之间的交互方式更加多样和复杂，软件协同逻辑与软件实体的计算逻辑开始呈现分离的趋势。在传统的结构化和面向对象软件技术中，与软件实体相比，协同处于从属地位，表达上被固定为过程调用或方法调用，分布上隐藏于计算逻辑之中，实现上则主要依靠程序设计语言或中间件提供的设施。而在互联网环境下，提供各种服务的软件实体都是高度构件化或服务化的，自主性更强，应用只能在尊重这种自主性的前提下协同它们，以达成应用目标；同时，为了适应用户需求和环境的变化，协同的方式必须灵活多样，可动态

调整；这不仅要求软件协同机制能够从软件实体中分离出来，而且应作为相对独立的机制加以研究与实现。例如，网构软件提出以软件体系结构中可定制的自定义连接子（Connector）来刻画自主构件之间的协同机制^[17]，服务计算中 WS-BPEL（Web Service Business Process Execution Language）和 WS-CDL（Web Service Choreography Description Language）等，都可以视作是协同机制的独立实现方式。

(2) 系统外部环境日益成为构造软件系统的基本要素之一，需要显式化的处理。外部环境的基本特征、构成成分和变化规律必定会影响软件的基本形态和运行支撑技术。随着软件技术的发展和变革、软件应用的不断深化和领域延伸，关于环境特征、环境建模以及环境信息处理等方面的工作近年来得到了重视并取得了初步进展。例如，环境感知（Context-Aware）机制是普适计算最为重要的支撑技术^[20]；网构软件提出环境驱动模型以支持软件系统自主感知并适应外部环境变化^[21]；基于 Web 的情境应用（Situational Applications）^[22]的开发也需要建立在充分感知并理解外部环境的变化的基础上。

此外，还需要指出的是，随着计算技术的应用越来越广泛，各类应用软件系统越来越复杂，人们越来越关心软件系统是否更好（更快捷、更安全、更可靠、更灵活）地解决了现实世界中的问题，为人们提供可信的服务。软件模型如何以简洁明了的方式表示除功能性和结构性之外的质量属性（如可靠性、可用性、安全性、高性能、适应性等），而这些表示又能被计算机系统所理解、并能方便快捷地转换成计算机软件的实现，也是软件模型值得研究的重要问题。

3.2 编程语言

编程语言作为软件模型的表示载体和描述工具，一直是计算机软件技术发展的主线之一^[15]。从最早的汇编语言到高级语言，到结构化语言（如 Fortran、Pascal、C 等），再到面向对象的语言（如 C++、Java 等），编程语言的抽象视角在不断变化，抽象级别也在不断提高。编程语言总是力图以更一致、更准确的方式实现从问题域到解空间的映射，更方便地表达问题求解思路，从而使开发者可以用更少的代码完成更多的工作，用更优的结构来控制编程复杂性，用更加严格规范的语法来预防编程错误等。换言之，编程语言的设计总是以更具表达能力和提高软件开发效率和质量为主要追求目标。

近年来，为了适应软件模型的变化，一些工作开始关注从较高层次来描述软件实体及其交互。例如，CORBA 构件的接口描述语言（Interface Description Language, IDL）、Web 服务描述语言（Web Service Description Language, WSDL）均不同程度地支持了软件实体的自描述，如操作、参数、传输协议等语法层面的规约，一些研究工作如 OWL-S（Ontology Web Language for Web Services）还加入了软件实体行为等语义层面的规约；体系结构描述语言（Architecture Description Language）、WS-BPEL、Agent 通信语言（Agent Communication Language）等则将软件实体之间的协同逻辑从计算逻辑中分离出来，作为相对独立的机制加以规约；Context Fabric^[23]、Gaia^[24]等关注对软件实体上下文环境（如时空信息、隐私、用户偏好等）的抽取和建模，等等。

需要指出的是，就编程语言自身而言，自面向对象之后，在过去三十年并无发生革命性的变化，大量的研究实践主要集中在编程语言的开发工具（如集成开发环境、调试器、重构等）、支撑框架（如.NET、Ruby on Rails 等）和程序库上，以及为开发者提供更多的编程辅助（如垃圾回收、类型安全检查等）等。当前，编程语言有三个发展趋势值得关注。

(1) 声明式（Declarative）的编程风格得到更多关注并大量应用。和命令式（Imperative）