软件适应性技术
——从个体适应到群体适应

丁 博 史殿习 著

科学出版社

软件适应性技术

——从个体适应到群体适应

丁 博 史殿习 著

科学出版社

北 京

内 容 简 介

本书从体系结构层面入手，研究适应性软件的构造技术。首先以较大篇幅对软件适应领域的已有研究进行较为全面的阐述和分析，在此基础上阐述作者近年来在个体适应和群体适应方面所取得的成果，包括融合个体适应性和群体适应性的软件自适应概念模型、自适应软件个体构造方法、集中决策的群体自适应机制、非集中决策的群体自适应机制等，以及基于这些成果形成的 Auxo 技术体系。

本书是对作者近年来在软件适应性技术领域工作的一次全面梳理和系统阐述，适合软件工程等相关领域的科研人员阅读参考。

图书在版编目（CIP）数据

软件适应性技术：从个体适应到群体适应/丁博，史殿习著. —北京：科学出版社，2013

ISBN 978-7-03-038785-1

Ⅰ. ①软… Ⅱ. ①丁… ②史… Ⅲ. ①软件适应 Ⅳ. ①TP311.5

中国版本图书馆CIP数据核字(2013)第236195号

责任编辑：张 濮 马晓晓 / 责任校对：宣 慧

责任印制：张 倩 / 封面设计：迷底书装

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

双青印刷厂 印刷

科学出版社发行 各地新华书店经销

*

2013 年 10 月第 一 版 开本：720 × 1 000 1/16

2013 年 10 月第一次印刷 印张：12 1/2

字数：252 000

定价：58.00 元

(如有印装质量问题，我社负责调换)

序

如何理解软件适应性？如何理解适应性软件？

简言之，软件适应性是指软件系统适应环境变化的特性，具备这种特性的软件就可以称为适应性软件。从这个意义上讲，适应性软件早已有之，发展到今天已相当普遍，而未来发展将更具挑战性。

在大型主机时代，当大型计算机逐步应用于金融服务和民用航空等关键业务领域的时候，如何适应硬件系统问题成为软件适应性在早期重点研究和解决的问题，具有容错特性的软件就是一种适应性软件。在个人计算机时代，如何适应不同用户的个性化需求成为个人计算机厂商吸引用户、争夺市场的关注点，软件适应性开始从适应硬件系统问题向适应用户个性需求转变，个人应用软件（如个人办公软件）中适应用户个性化的特性大量出现。进入网络时代后，软件适应性越来越关注环境属性、系统属性和用户属性的完美融合。例如，今天的移动互联网用户几乎不用关心是通过 3G 还是通过 Wi-Fi 接入互联网，因为智能移动终端上的软件能够根据信号变化自主选择接入协议，越来越多的普通用户已经在不知不觉中享受着适应性软件带来的好处。

软件适应性的提高有力地推动了信息技术的快速普及。同时，伴随着信息技术的广泛应用，人们对软件适应性的要求越来越高，导致适应性软件的发展面临诸多瓶颈。

一方面，适应性软件所处的环境越来越复杂。早期适应性软件关注的环境仅限于其运行的硬件平台，需要适应的环境状态往往是有限的、可预期的，适应状态变化的策略通常是明确的、可预设的，适应性问题一般还可以转化为最优化问题。当适应性软件开始关注五花八门的用户个性需求、关注所作用的丰富多彩的现实世界时，环境状态不再是有限的和可预期的，适应状态变化的策略不再是明确的和可预设的，适应性问题成了尽力而为问题。

另一方面，适应性软件本身越来越复杂。复杂理论和非线性科学的先驱 John Holland 教授有一句名言：“适应性造就复杂性。”软件适应性的提升导致适应性软件系统越来越复杂，已经呈现复杂系统的若干特征。从本质上来说，适应性软件系统的适应与社会、经济和生物系统的适应一样，都是复杂系统在其外部环境发生变化时，通过内部补偿性的调整来应对变化的过程。作为一门只有几十年历史的软件学科，人们对适应性软件系统的构造、运行和演化的独特规律远没有很好地理解和把握。

本书是国防科学技术大学分布计算技术与系统研究团队近年来努力突破适应性软件发展瓶颈的新成果。我与本书两位作者围绕适应性软件问题进行了长期的合作研究。丁博博士曾在我指导下完成了分布对象中间件技术的硕士学位论文和软件适应性技术的博士学位论文，本书的主体内容就是基于他的研究成果。我和史殿习博士在分布计算技术与系统领域并肩工作已有二十年时光，本书的部分内容源自他主持完成的“面向普适计算的自适应软件平台”863 重点课题的成果。

本书的内容有以下三个特点。

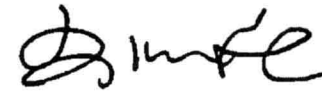
一是从软件体系结构切入。软件适应性的研究大多沿着人工智能的线索展开，而本书则从软件体系结构出发研究适应性软件。本书系统分析了软件适应性技术的发展，借鉴现实世界中生物、社会和经济系统所具有的适应性结构或表现出的适应性，将其规律与软件本身的特点结合，反过来指导复杂软件系统的构造与演化。

二是将人的适应能力纳入其中。软件适应性的研究大多强调“自适应”，但难以应对超出预期的环境问题。“人在系统中”是本书适应性软件构造的一个重要特点。我们认为，需要通过人与适应性软件的结合来实现软件适应能力。特别是在支持人对软件适应能力进行在线调整的方法和基础设施、多个个体如何形成具备适应能力的群体等方面，本书具有创新性。

三是强调研究工作的实践价值。本书坚持从具体问题入手，通过案例理解软件适应性问题，也通过案例检验软件适应性技术的有效性。本书还坚持一种软件价值观：可运行才能降生，可应用才算成熟。本书的核心成果均有可运行的软件系统支撑。因此，相信本书的成果对本领域的研究者和实践者都会有较大的启发。

在人围着计算机转的年代，适应性软件关注的首先是尽可能把计算机系统管理人员解放出来；当计算机围着人转的时候，适应性软件必然关注普通用户的需求。在人围着网络转的时候，适应性软件只是力求解放网络系统管理人员；当网络随世界而动时，适应性软件必然关注人-机-物融合的全新世界。

广义地讲，适应性软件是一种理想，其发展是无止境的。希望本书能够成为留在追求理想的成功道路上清晰可见的足迹。



国防科学技术大学

教育部长江学者特聘教授

2013 年 9 月

前　言

2011 年 4 月 21 日，云计算提供商 Amazon 在美国弗吉尼亚州北部的云计算数据中心发生严重宕机事件，很多基于 Amazon EC2 构建的热门网站都不能响应用户访问请求（如 Foursquare、Quora 和 Heroku 等）。但是在此次灾难中，付费电影点播网站 Netflix 却成为了幸存者。虽然在宕机期间发生影音传输变慢、用户抱怨大幅增加的现象，但并没有像其他网站一样服务完全中断。

Netflix 之所以幸免于难，是因为其在将服务从企业内部数据中心向 Amazon 云端迁移时，并未假定系统的未来运行环境是静态不变的，硬件失效、软件故障、外部资源缺失等现象在设计时就被认为是常态化。因此，系统内置了相应的检测和调整机制：系统能够在短时间内检测到故障和失效，并立即在其他地方（如另一个可用域中）启动备份服务；在备份机制生效之前，或者问题无法在短时间内解决时，系统仅仅是缓慢地降低可用性，而不是完全崩溃。Netflix 甚至专门设计了一个称为 Chaos Monkey 的常驻进程，它的作用就是随机将系统中的各种进程删除，以测试系统的存活能力。由于这种在线检测和调整能力的存在，当 4 月 21 日 Amazon EC2 基础设施发生故障时，Netflix 相关的服务被及时迁移到 Amazon 未发生故障的区域，系统通过降级运行保证了服务的持续性。

Netflix 的这个例子说明了软件适应的重要性：随着云服务、Cyber-Physical 系统、移动计算等大型分布式系统的出现，软件本身变得更复杂、运行环境变得日益开放动态，需要能够根据环境变化和运行状态，对软件进行动态调整，以保证其能够持续、高质量地提供服务。

软件适应涉及环境感知、适应决策、在线调整等环节，是对软件能力的一种理想期望。研究者针对软件适应这一目标已经开展了大量工作。例如，早期程序语言反射研究的关注点就是如何在运行时感知软件状态并调整其行为[1-2]；与自适应密切相关的自主计算概念强调软件系统的自配置、自优化、自修复和自保护等；Rainbow[3]、K-Component[4]、MADAM[5]、网构软件技术体系[6]等强调基于软件体系结构技术实现自适应；Hadas[7]、Accord[8]、PCOM[9]、ACEEL[10]、CompAA[11]等从构件模型角度研究如何构造具备适应性的软件；OpenCOM&OpenORB[12]、Gaia[13]、CASA[14]及早期的 Quo[15]、DynamicTAO[16]等则从中间件/软件框架角度支持或实现软件适应；透明计算[17-18]从计算系统架构的角度为实现软件适应提供了新的思路。此外，具备自管理能力的软件体系结构描述语言和框架[19]、20 世纪 90 年代早期提出的适应性程序概念[20]、分布式环境下负载均衡和容错等大量研究

均可划归软件适应的范畴。

然而，我们距离“以系统化方式构造适应性软件”这一目标仍然很远，特别是构造大规模的、具备适应性的复杂软件系统。在软件工程实践中，一些软件的适应能力仍完全以 ad hoc 方式设计和实现，程序开发者甚至要重新发明软件适应过程的“轮子”。许多原因造成了这一现象，包括：现有研究多数聚焦于适应机制，仍缺乏系统化的软件工程方法和工具；运行环境的不确定性使得在开发阶段很难预先“设计”软件的适应能力；分布环境下的软件适应过程有其独特特点，例如多个自主结点需要协同适应；等等。

本书从软件体系结构层面入手，对适应性软件的构造技术展开探索。我们将首先以较大篇幅对软件适应领域的已有研究进行阐述和分析。在此基础上，系统化地阐述作者近年来在软件自适应领域所取得的成果，包括融合个体和群体适应性的软件自适应概念模型、自适应软件个体构造方法、集中决策的群体自适应机制、非集中决策的群体自适应机制等，这些成果形成了名为 Auxo 的技术体系。虽然这一技术体系尚有许多需要进一步完善的地方，但希望能够对本领域的研究工作有所贡献。

本书分为两个部分。

第一部分是软件适应基础理论和研究现状分析，介绍软件适应的基本概念、使能技术、典型实践和前沿探索。

第 1 章介绍软件适应的概念，给出软件适应研究背景和典型应用场景，阐述软件适应与软件演化的关系，以及软件适应的系统科学基础。

第 2 章以“感知-决策-执行”软件适应基本周期为主线介绍软件适应活动特征分类，以及各个环节的使能技术。

第 3 章介绍软件适应典型实践，对国内外与软件适应密切相关、具有较广泛认知度的软件工程研究项目进行阐述和分析。

第 4 章在对典型实践进行总结的基础上，给出本领域的未来研究挑战，概述研究者近年来针对这些挑战所展开的初步探索。

第二部分介绍我们所开展的名为 Auxo 的自适应软件构造方法和框架实践。

第 5 章给出 Auxo 技术体系概述，提出指导我们工作的 Auxo 软件自适应概念模型，给出该概念模型参考实现的基本架构。

第 6 章提出基于控制理论的软件个体复合控制过程，在此基础上给出由 Auxo 构件模型、Auxo 单元框架、软件自适应能力在线调整方法和 AuxoDL 语言组成的自适应软件个体构造方法。

第 7 章阐述 Auxo 参考实现中集中决策的群体自适应机制，包括任务规约驱动的群体动态聚合方法、跨单元连接子的实例化方法等。

第 8 章阐述非集中决策群体自适应的分布式约束优化建模方法，提出针对低

约束密度问题的算法，并对算法进行实验验证。

第 9 章给出基于 Auxo 参考实现的自适应中间件总体设计，给出若干实验和测试结果，简要介绍基于该中间件的第三方应用，展望未来工作。

本书的写作和出版得到了国家 863 项目（2011AA01A202）和国家自然科学基金项目（9118008、61202117）等的支持。在成稿过程中，国防科学技术大学分布计算技术与系统课题组的老师和同学提出了许多有价值的建议。此外，本书第 9 章所提及的“面向普适计算的自适应中间件 UbiStar”原型系统是在国家 863 计划课题（2006AA01Z198）支持下，由国防科学技术大学、清华大学、浙江大学和西北工业大学多方合作的结晶，在此谨一并表示衷心的感谢。

本书是对作者前期工作的小结，由于水平有限，书中不妥和疏漏之处难免，恳请读者批评指正。

作　者

2013 年 6 月

目　录

第二部分　研究实践篇

第一部分 发展分析篇

第 1 章　软件适应基本概念

In the struggle for survival, the fittest win out at the expense of their rivals because they succeed in ***adapting*** *themselves best to their environment.*

——达尔文（1809—1882），《物种起源》，1859

适应是指某一对象在其环境发生变化时，自身有针对性地进行调整以应对变化的过程。具体到软件技术领域，适应代表了人们对软件能力的一种理想期望——随着软件在社会经济和生产生活中扮演的角色越来越重要，它应当能够像生物体、社会经济系统等一样，具备灵活、主动应对环境变化的能力。这一目标的实现涉及控制理论、软件工程、人工智能和复杂系统等多个学科，无论对研究者还是实践者都是巨大的挑战。

1.1　软件适应的定义

自 20 世纪 90 年代以来，学术界和工业界从桌面软件、实时嵌入式系统和企业级系统等不同背景出发，对软件“适应”和“自适应”的内涵进行了研究，相关定义如表 1.1 所示。

表 1.1　软件适应的相关定义

作者（年份）	定　义
Bihari（1991）	软件适应是指改变软件可靠性或时间相关属性（timeliness），而不影响软件功能其他方面的软件修改活动[21]
DARPA（1997）	自适应软件可以评估其自身行为，并在评估结果指明其无法完成指定任务或有可能实现更佳的功能或性能时，改变自身的行为[22]
Oreizy（1999）	自适应软件修改自身行为以响应其操作环境中的变化，此处操作环境包括了任何软件系统所能观察到的东西，如用户输入、外部硬件设备和传感器、程序插装（program instrumentation）等[23]
Chen（2001）	自适应软件可以依据执行环境和用户需求的变化而改变其行为[24]
Taylor（2009）	动态适应（dynamic adaptation）是指软件在运行时系统功能变化的能力，在这一过程中无需系统重启或重载入[25]
Salehie（2009）	自适应软件通过调整其部件和属性来响应软件系统自身和环境的变化[26]

本节将从软件适应若干案例入手，通过案例分析，结合前人所给的定义和实际应用需求给出软件适应和软件适应性的概念，进而给出个体和群体适应的概念。

1.1.1　软件适应案例

一些具有代表性的软件适应案例如下。

场景 1.1（多链路通信中间件）　在服务器上配备了互为备份的多块网卡（和多个接入链路）。在链路发生故障时，服务器上的中间件可手动或自动切换链路，使得上层的网络应用可以提供 7×24 h 不间断服务。多链路通信中间件如图 1.1 所示。

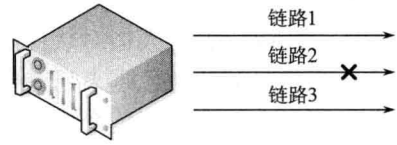


图 1.1　多链路通信中间件

在上述场景中，当监测到软件正在使用的链路发生了故障时，通过调整中间件内部的通信层配置来使用其他的通信链路，以补偿环境变化，此时称软件表现出了环境适应能力。

场景 1.2（Cyber-Physical 系统中的灾难管理）　某建筑内部部署了软件密集型安防系统，其功能之一是检测火灾并在必要时启动应急响应流程（如开启相应区域的喷淋阀门）。然而，某些设备、线路可能会在火灾中损坏，软件系统需要通过重配置来进行补偿，例如，在线路损坏时，传感器主动开启无线工作模式。

在上述场景中，适应的目标是保证安防系统的功能正常发挥，而适应的手段是软件系统的重配置。

场景 1.3（云计算中的自适应服务器池）　云计算基础设施可以通过虚拟机等手段动态申请和释放计算资源。某个云服务后端采用服务器池来处理用户请求。为了使运行费用最小化，系统能够依据当前负载调整服务器池的大小。云计算中的自适应服务器池如图 1.2 所示。

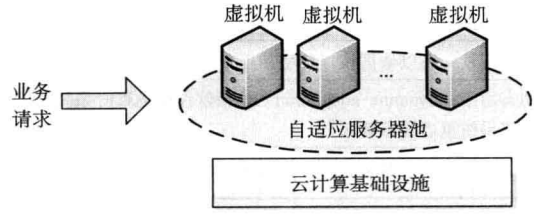


图 1.2　自适应服务器池