网络软件工程技术的发展及其对策*)

The Position and Countermeasure on Network Software Engineering

杨 云 徐永红 张 琨 刘凤玉

(南京理工大学计算机系 南京 210094)

Abstract This paper discusses the development of software engineering on net era. Not only does it make some important account on software's character, but also it gives good resolvent in designing network software.

Keywords Software engineering, Software component, Role technology, Network security

1 软件工程的发展

软件开发从初期小规模程序发展到现在的结构复杂的程序体结构,并且可以自动生成软件。各种类型的大型软件系统(如操作系统)成为了一种"智能产品",软件开发也逐步模型化和标准化,从软件设计(COCOMO)到软件工程(CMM),直到ISO-9000(ISO-10006)。随着网络在世界范围内普及,不断日益增大的需求,对软件系统提出了更高要求,使得软件系统设计的复杂度越来越高,程序量越来越大,软件的可靠性越来越显得重要。另一方面硬件技术的数量级更新,促使软件技术必须持续革新和发展,基于新的软件设计理论、语言和软件设计方法的出现,推动了整个计算机科学和技术的发展。

(1)面向综合化和专业化的现代软件工程

初期软件开发是以程序为中心的。以60年代由于软件规模扩大而产生的"软件危机"为契机而出现了"软件工程",之后各式各样的技术和构想在程序设计中得到应用,取得了很好的效果。依靠这些技术建立的现代软件工程,使得现代软件工程具有面向综合工程和特定的应用软件包的特点。

①*复合型现代软件工程* 在开发现代大规模复杂软件时,要求开发者在软件开发过程中使用方法论规则对问题进行计划、需求分析、设计、程序编码、测试和运行维护,能够开发和利用适用于软件和硬件两方面环境变化以及用户新的需求的对象软件包。现代软件工程不仅仅是进行程序设计,而且要求开发者具有多方面知识(政治的、经济的和社会的),按工程化原则和方法组织软件开发、综合运用各方面知识和技术的复合型工程。

②*面向对象的软件工程* 软件被广泛应用于各种产业和社会领域的各个方面,一般把这种领域称之为应用对象或简称为对象。对象具有其特有的要求和问题结构,软件开发中也非常重视这种能够反映对象特性、面向对象的软硬件基本构造和面向对象开发方法论的研究。例如,组合软件的reactype软件就是其中的一种,和外部的相互作用是其本质的要求,因此"基于状态转移模型"特有的开发方法论,目前被广泛应用。

(2)现代软件工程的发展

①*软件过程的发展* 过程的概念是从1970年Royce提出的"瀑布型过程"开始的。80年代由于GUI技术的发展,人们提出了"智能化和简单化"设计问题,因此,产生了被称之为"原型"的新的开发过程。进一步,1987年基于"过程程序控制"的过程设计方法论产生了,这种方法论和能力成熟度模型CMM (Capability Maturity Model)等进一步发展了过程的组织改善技术,使得过程概念到了90年代迅速得到普及。在开发论方面,"面向对象程序设计方法"和"再利用技术"一起构成了软件开发的基础技术。

另一方面,由于Internet的发展,把时间作为基础的竞争模型,软件的提供和生命周期的概念也发生了变化。它促进了把适应要求的阶段开发、进化的自然智能过程和时间作为基础过程模型的研究发展。为了使软件能够适应硬件和用户不断变化的要求,软件的发展也是非常必要的。另一方面,智能开发过程也是适应软件发展的一种重要途径。

②*软件产品的发展* 产品开发,已经从单细胞构造进化为根据功能、单体机器组合而达到高效率的OA机群。从60年代的中期的OS/360操作系统的出现,软件就开始了分化。80年代的开放化和规模的小型化,应用型软件从二层进一步分化为三层的顾客/用户服务。90年代有三个大的变革,软件的"机能构造"和"面向角色程序设计"等高水平结构化技术出现,产生了面向Internet的软件机能构造以及软件服务化和大容量。

面向对象程序设计方法,是基于容器化、逻辑上高独立性、程序产品组合和在上层结构进行分析、设计的。但是为了设计出大规模、复杂度高的软件,必须有一种更高水准的结构化技术。因此,提出了一种能够对软件的所有结构进行设计的"软件机能构造"技术(即面向角色程序设计技术),进一步为了把"软件机能构造"作为在实际应用中可操作的软件设计方法,又提出了基于"软件机能构造"的开发体系和局部设计模型。基于此,把对象群作为角色、将各种角色进行拼装组合的技术,目前正越来越受到软件设计者的关注。

Internet的发展也带动了计算机软件、硬件的不断更新和升级。从仅供特定的顾客和用户使用的顾客/用户服务系统,发展到对任何顾客和用户都开放的网络系统,电子商务等各种各样新软件和服务得到了开发和应用。

另外,在Internet上,软件、需求和服务还在进一步融合,网络软件ASP(Application Service Provider)等有可能成为主流产品。现在,用户没有必要拥有所有的软件产品,网络上所提供的软件服务可以免费下载。

*)本文研究得到国防科工委应用基础基金(NO. J1300D004)资助。杨 云 副教授,博士研究生,主要研究领域为软件工程、网络安全。徐永红 博士研究生,主要研究领域为网络安全。张 琨 博士研究生,主要研究领域为网络信息安全。刘凤玉 教授,博士生导师,主要研究领域为信息安全理论与技术、多媒体计算机技术。

③软件管理和组织的发展　在管理方面，已经不是一个人担当所有软件开发过程，而是根据技术分工和过程控制要求，由各种专家担任。如域名工程师、系统工程师和进行 component 开发的工程师等。还有，用于进行企业战略决策支持的 CIO(Chief information Officer)或 CTO(Chief Technology Officer)其作用也越来越受到重视。这些，使得企业的组织结构和人才培养发生了变革，促使产业结构分化。

另一方面，基于 Internet，依靠多个组织，采用国际分工或开放式的组织结构，使得信息共享型开发等软件，在世界范围内成为可能。

2　网络时代软件业面临的挑战

Internet 既是一个巨大的计算机，又是一个巨大的软件，它形成了一个虚拟企业或一个虚拟社会，在带来新的软件开发、服务机会的同时，又产生了新的研究课题。因此，必须寻求建立与 Internet 时代相适应的软件工程新体系。

(1)网络化和大众化编译软件对软件工程提出的课题

①网络化所带来的课题　由于网络化，软件由点向面广泛展开。如在电子商务等新的软件的开发中，一般有下列要求：A)在网络上，综合了多种软件、能够提供各种服务；B)与动态变化网络或事务性模型相适应，能够与变化了的环境同步的软件；C)能够提供 24 小时 365 天服务，高信赖性、高完备性和高安全性；D)适应用户需求扩大，和规模、复杂度的加大。

为了满足这种要求，有必要建立新的软件开发、开发过程和开发方法。进一步，由 Internet 引发的软件和服务开发，也带来了跨越国境的世界范围内的竞争。我国软件产业才刚刚建立，尚未形成具有国际竞争力的规模。特别是我国的软件产业，必须进行结构性调整，紧跟世界潮流，由生产专用软件(根据用户的要求设计的软件产品)，转到建立包含有能够生产供特定业务使用、服务的软件开发的新的事务性模型的开发中心。

②大众化编译软件开发课题　编译软件，即使在今天，也被认为是硬件的一部分，依赖于人海战术开发。但是，移动电话软件规模如果超过 1M LOC，编译软件也变得规模化和复杂化。因此，有必要使被称之为“编译软件工程”的技术体系化。大众化编译软件对一般用户来说，是在无意识地使用，应该建立把以计算机为中心的设计，转移到以人为中心、以用户为中心的设计中来。

③网络安全性问题　现在是最重要的问题，近年频繁发生的黑客攻击事件中，黑客使用最多的一种攻击手段是 DDOS(Distributed Denial of Service)，它使被攻击网站不堪重负，不能提供正常服务。而通常的 TCP 基于窗口的端对端的拥塞控制，随着网络规模的扩大和结构越来越复杂，具有很大的局限性。同时，IP 层对流量的控制功能也很弱。一方面，对 TCP/IP 协议进一步完善，另一方面，用于网络安全的加解密技术、防火墙技术、安全路由器以及入侵检测技术、流量控制和负载均衡技术研究必须进一步深入。另外为了保证软件本身的安全性，要扩大对软件检测技术的研究。对软件进行完全自动检测在技术、成本和时间上几乎是不可能的，但可通过对软件各部分模块自动检测，以达到对整个软件系统的检测，从而提高软件产品的可靠性。

(2)超软件工程(Extreme Software Engineering)和软件工程的基础技术

通常超高层大楼的建筑，称为“极限工程”。随着超高层建筑越来越高层化，用于世界金融网络或航空管制的软件的规模也越来越大，复杂度也越来越高。80 年代到 90 年代开发的美国航空管制软件破绽的暴露，使人们认识到超巨大软件开发的风险仍很高，面向对象的开发技术有相当的局限性。

另一方面，在软件工程的基础技术方面，其形式化、检测、试验技术等存在不少技术上的问题。因此，必须确立软件工程的基础技术。

3　网络软件工程技术

(1)软件进化(Software Evolution)技术

为了使软件能够满足用户需求和适应环境变化，作为一种软件开发技术，软件进化是必需的。目前，大规模软件的开发中，进化已成为主要课题。作为软件进化经验准则，主要包括以下 3 点：

①第 1 法则　连续变化：为了使软件能够被继续使用，软件变化是不可避免的。

②第 2 法则　复杂度的增大：伴随着需求的扩大，软件的复杂度也随之增大，必须努力抑制复杂度增大。

③进化的基本法则　软件的变更量依据母体软件的规模决定。

基于再生工程技术、仿生工程技术软件，如美国的 EDCS (Evolutionary Design of Complex Software)等，把软件进化作为主要内容纳入它的研究计划。世界各国如此重视软件进化，主要基于下述 3 个理由：

①进化的必要性：基于 Internet 的软件动态环境和事务、社会的急速变化，带来了对软件需求的巨大变化，能够适应这种变化的软件进化，成为软件设计的主要课题。

②软件开发的危险性增大：由于软件规模扩大，基于 big hang 单一生命周期开发的危险性很大。为了降低危险性，采用使大规模软件具有阶段性进化功能的开发方法进行软件开发。

③基于开发现场保守问题：软件开发现场“保守”的主要内容是，能够反复对软件追加新功能。由于软件规模扩大和开发成本提高，基于母体软件进行再开发是不恰当的，因此，必须从“保守”软件转到“进化”软件。

作为进化型软件开发技术，目前，对它的研究主要有以下成果：①可能进化软件结构模型；②面向问题的可能进化软件结构模型；③动态适应环境变化软件构造论；④可能进化软件说明记述法；⑤可能进化软件构造支援论。

(2)软件构件技术

软件构件化技术，就是应用开发人员利用现成的软件构件装配成适用于不同领域、功能各异的应用系统。其目的是彻底改变软件生产方式，从根本上提高软件生产效率和质量，提高开发大型软件系统尤其是商用系统的成功率。软件构件至今没有一个严格的定义。目前比较典型的有四种说法，分别从不同角度对软件构件加以了说明：

①软件构件是一个并非不重要、相对独立而又能够被替换的系统中的一部分。

②处在运行状态的软件构件是一个将若干个程序动态链接在一起的程序包。

③软件构件是可以独立使用，并且不是由系统开发人员和系统最终用户而是由第三方提供。

④软件构件(商用构件)指的是一个能够自动化的商业过

程的软件实施办法。

根据上述定义，软件构件通常可分为白匣子、灰匣子和黑匣子三类。所谓白匣子是指在提供软件构件的同时也提供实现构件的全部源代码，在应用这个构件时，开发人员需要对源代码进行某些修改，然后才能将它集成到系统中实现一定的应用目的。灰匣子只提供有关界面部分源代码，开发人员在应用构件时对构件内核是不清楚的，只能在接口界面上做一些用户化工作。黑匣子则完全不提供源代码，只提供构件二进制可执行形式。

软件构件近年来的一个趋势是开发现成构件(Commercial Off the Shelf，简称为COTS)。其特点是：1)不是由应用开发人员而是由第三方做的，是现成买来用的，即强调构件的预制性；2)不提供源代码，强调构件封装性和透明性；3)强调可互换性；4)不是为某一个客户定制，而是在某一领域中通用。

总而言之，现成构件是具有预制性、封装性、透明性、互换性、通用性的一个软件单元。对于构件开发人员来说，关键是如何定义一个新构件，即如何在应用领域的应用模型中找出有共性、可通用部分做成软件构件。

因此在做构件之前，首先要明确它将适用于哪个应用领域，然后再根据这个领域的知识和应用模型抽出最合理构件定义。第二，构件相互之间的集成和装配。为了能够用构件集成应用系统，使构件能够装配互换，必须要有一个统一标准，如跨平台开放标准CORBA(公共对象调用中介结构)等。第三，实现构件之间相互通讯、如何实现数据共享。第四，新的构件化系统继承非构件化系统数据。

Java促进了软件构件化技术的开放与标准化。Java对软件构件化技术的促进主要是通过Java Bean，即通过做Java构件来实现的。用Java做出来的构件是跨平台的，直接支持网络计算，并且还可以实现更高水平的安全。

(3)基于角色的程序设计方法

程序设计方法已从面向数据流、面向数据结构和面向过程的程序设计方法过渡到面向对象的程序设计方法，目前正在向面向角色的程序设计方法过渡。

角色抽象了特定语境中实体的状态和行为。基于角色技术进行软件开发，不仅使软件系统能适应于当前语境，而且以可预见的代价适应于将来可能变化的语境中。角色技术以对象式方法为基础，以统一建模语言UML为规范，可以较好地解决大型复杂系统中传统方法难以解决的三方面问题：需求建模、访问控制和设计模式。

①角色技术是对象式方法的重要发展和完善。传统对象方法用类抽象对象的完整状态和行为，并负责创建对象，侧重于对象的设计和实现。角色也是对象的一种抽象，它抽象对象与特定语境相关的状态和行为，一个对象在其生命期中的不同语境中可扮演多个角色。角色间也有继承、关联等静态关系。角色技术注重对象的语境，角色建模适合于系统分析的建模，在类的设计中，将实现对象所扮演的角色。若对象分析和建模中忽略了语境和角色，类设计中将难以协调完整地实现对象特征，且可能隐藏问题。角色技术是在对象方法基础上发展和完善。尤其大型系统中对象特征复杂化，角色技术方显其重要意义。

②角色技术的各部分各有侧重且相互协调。需求建模侧重于业务建模和需求分析，角色用于抽象各种职责用户、业务对象、用例、工作流等，目的是以规范而自然的方法描述一个即将开发的系统，使开发人员和领域专家之间能建立有效沟通，而且指导软件开发。访问控制侧重于安全性设计，角色描述机构中团体和人员的权限和职责，目的是提供一种通用而高效的访问控制机制。设计模式侧重于在特定语境中对常见设计问题的解决，角色用于抽象特定语境中的对象，目的是提供一种简单而自然的模式描述方法，并描述一组常用模式，便于模式的理解、交流和重用，以提高开发效率。三种之间相互协调，协调基础有三：对象方法、角色原理、UML规范。

③角色技术是大型复杂系统开发的有效手段。小型系统的简单性表现在对象语境相对稳定，且开发时可预见其变化范围。大型系统的复杂性很大程度来源同一对象要适用于不同语境中，且系统开发时难以预见将来可能的语境变化，这要求软件系统不仅能适用于当前语境中，而且要适用于将来可能变化的语境中，且使维护代价限定在预见的范围之内。角色技术提供完整的解决方法。角色反映语境特征，角色建模可协调不同语境中的状态的行为，基于角色的访问控制使系统能适应人员频繁变化和机构重组，Role模式使当前开发的系统能以可预见的代价适用于将来可能变化的语境中。角色技术提供了处理大型系统复杂性的有效手段。

(4)软件检测技术

基本的软件检测方法主要有：模型检测、基于推理机理论的检测和依据定理证明的检测等，每种检测方法都有其固有的特征。黑盒法主要有：等价划分法、边界值分析法、因果图法、错误推测法；白盒法主要有：语句覆盖法、判定条件覆盖法、条件覆盖法、判定条件/条件覆盖法、多重条件覆盖法。采用工具的验证方法主要有以下三种：静态验证法、动态验证法和符号验证法。使用静态验证法和符号验证法这两种方法进行验证时不运行程序，而采用动态验证法时要运行程序，需要其过程和结果。符号验证法是允许对测试数据x使用符号方法，对作为符号而给出的x，它能够以符号方式得到测试输出P(x)(P为目标程序)。这种方法的优点是能够得到三者当中与区域D最近的测试集合T(可靠测试集合)。目前这种方法正在研究之中。

随着计算机网络化和大众化，由计算机网络所产生的社会环境正在影响着人们生活，为了实现软件带给人们欢乐和幸福的环境，人们正在不断地在网络上推出各种服务。当今的时代，已经从计算机转到网络、从计算机硬件转到计算机软件，对软件的新研究、开发和事务性服务充满着机遇和挑战。

参考文献

1 Anton A I, Potts C. Requirements Engineering in the Long-term Fifty Years of Telephony Feature Evolution. Proc FEAST 2000, Jul. 2000

2 Aoyama M. Agile Software Process and Its Experience. In: Proc. of 20th ICSE (Int's Conf. on Software Engineering), 1998

3 青山幹雄. ソフトウェアパターンのモデル化とパターン进化のパターン. 情报处理学会ソフトウェ工学研究会, 1999

4 片山卓也. 进化发展するソフトウェアの原理. 情报处理, 1999

5 郑人杰. 软件工程(高级). 清华大学出版社, 1999