Internet 计算环境下的新型软件形态——网构软件(Internetware)综述

薛云皎　徐如志　钱乐秋
(复旦大学计算机与信息技术系,上海 200433)
E-mail:yjxue@fudan.edu.cn

摘　要　软件形态受到应用、平台和技术发展的影响而不断演化。由于 Internet 这一新的计算环境的普及,传统的软件形态逐渐无法适应 Internet 环境下的开发和应用,需要研究新的软件形态和软件技术。该文在分析软件发展历史的基础上结合相关研究说明了基于 Internet 环境的网构软件(Internetware)是未来软件的发展趋势,介绍了网构软件的基本特征,并说明了网构软件领域的研究问题。

关键词　Internet　软件　形态　网构软件　演化

文章编号　1002-8331-(2004)14-0038-03　**文献标识码**　A　**中图分类号**　TP311

A Summary of Internetware – a New Software Modality under the Internet Computing Environment

Xue Yunjiao　Xu Ruzhi　Qian Leqiu
(Department of Computer and Information Technology, Fudan University, Shanghai 200433)

Abstract: The software modality keeps evolving under the influence of application, platform, and technology development. The traditional software modality has got unfit for the developing and application under the Internet environment because of the popularization of Internet. New software modality and technology need to be researched. This paper explains that the Internetware based on Internet is the future trend of software's evolvement, introduces its essential feature, and illuminates several research problems in this field.

Keywords: Internet, software, modality, Internetware, evolvement

1　软件形态的演化

软件是在计算机硬件设备的基础上,借助用户控制、驱动计算机进行数据处理以解决实际问题的逻辑手段,是计算机的灵魂。目前,对于软件的组成成分,普遍接受的定义是:

软件(Software)是计算机系统中与硬件(Hardware)相互依存的另一部分,它包括程序(Program)、相关数据(Data)及其说明文档(Document)[1]。

与此定义相关,软件如何开发(Development)、如何存储和访问数据(Data)、如何与硬件(Hardware)交互、组成软件的各部分如何通信协作(Collaboration)、软件如何运行(Run)以执行功能联合起来构成了软件的形态。随着计算机技术的发展和计算机应用的不断深入,软件形态经历了五个主要的演化阶段[2]。

第一代(50~60 年代),是以 Algol、Fortran 等编程语言为标志的算法技术。程序设计以硬件体系结构、数据结构和算法为中心,软件生产率低,程序编写复杂,软件设计上个人发挥较多,给软件的修改、维护带来极大的困难,导致了 60 年代末的"软件危机"。

第二代(70 年代),是以 Pascal、Cobol 等编程语言为标志的结构化软件技术[3]。这种技术以功能为中心,采用自顶向下逐步求精的设计方法和单入口单出口的结构化控制结构,大大改善了程序的可读性。伴随着结构化软件技术而出现的软件工程方法(包括 CASE 工具),使软件开发的范围从只考虑程序的编写扩展到从软件定义、分析、设计、编码、测试到使用、维护等整个软件生命周期。软件不仅仅是程序,还包括开发、使用、维护程序所需要的所有文档,使编程工作在整个过程中所占的比重大大降低。结构化软件技术使软件由个人的发挥变为团队的工程产品,大大改善了软件的质量与可维护性,但软件开发的成本却大大增加了。

第一代和第二代软件开发都是以问题为中心,强调从需求中标识实体,通过结构分解将功能分布到多个实体中,这使得功能抽象较为困难,分解随意性强,当需求发生变化时难以通过仅改变实体来满足变化,常常导致软件结构的变动,致使软件可维护性和适应性较低[4]。

第三代(80 年代)是以 Smalltalk、C++等为代表的面向对象技术(Object-Oriented)。

OO 思想发端于 Nygaard Kristen 的 Simula[5]。OO 以对象作为最基本的元素,它将软件系统看成是离散的对象的集合。一个对象既包括数据结构,也包括行为。一般情况下,一个对象与

基金项目:国家 863 高技术研究发展课题"基于 Internet 以构件库为核心的软件开发平台"(编号:2001AA110241)资助
作者简介:薛云皎,博士研究生,研究方向:软件工程,软件复用,构件库管理系统,基于构件、构架的软件开发方法。徐如志,博士研究生,研究方向:软件工程,软件复用,构件库管理系统,构件生产与组装。钱乐秋,教授、博士生导师,研究方向:软件工程,软件复用构件库管理系统,构件生产与组装,软件测试,基于构件、构架的软件开发方法。

现实世界的一个事物相对应[6]。OO 方法以对象为分析问题的中心,以类及其继承作为程序的构造机制。对象技术的最大优点是帮助分析者、设计者及用户清楚地表述概念,互相进行交流,并作为描述、分析和建立软件文档的一种手段。对象的概念极大地提高了软件的易理解性和可维护性,进一步地又使得从软件分析到软件设计的转变非常自然,因而有效地降低了软件开发成本。相反,算法技术和结构化技术中从分析到设计的转变就缺乏这种自然性,因为这两者的实现基础是计算机指令系统,而不是人思维中的概念。此外,OO 技术中的继承、封装、多态性等机制,直接为软件复用提供了进一步的支持。OO 技术开辟了通过有效的软件复用来提高软件生产率的新途径。

但随着应用复杂度的不断提高,软件系统规模不断扩大,对象技术也逐渐不能适应新的需求,其原因在于对象粒度过小,无法表达高层的应用抽象,同时被动的对象也缺乏对环境的适应性,尤其是无法应付网络环境下的协作需求。

第四代(90 年代)的软件技术是以 CORBA 等为代表的分布式面向对象技术(Distribute Object-Oriented)[7]。

进入 20 世纪 90 年代以来,随着网络技术的发展,计算机应用迅速从单机扩展到网络。充分利用网络中的计算资源、通过协同通信提供计算服务成为急迫的要求。同时,网络体系结构、网络协议和网络操作系统的发展愈发呈现多样化的趋势,使得异构环境下分布式软件的开发成为一种主流需求。OO 技术对软件的复用,仅限于单台计算机上、同种操作系统与编程语言环境下的软件复用,对象往往仅存在于一个程序中,外部程序无法感知和访问这些对象。异构环境分布式系统中的软件复用,要求能够复用不同计算机上、不同操作系统或语言环境下,由不同人员不同时间开发的软件模块。具体地,就是要解决不同软件之间的组合性、互操作性、可移植性等技术问题。

在 DOO 得到快速发展以前,为提供网络应用解决方案,曾出现过客户/服务器(Client/Server)模型,能够在客户之间有效地实现服务器资源的共享[8]。但由于 C/S 体系结构中资源需求集中在服务器上,不能完全支持使用分布资源的应用,而且只适应于单一固定的应用,因而使用范围极其有限。

DOO 技术采用面向对象的多层客户/服务器计算模型[9],在系统软件与应用软件之间提供一个统一的软总线(Software Bus)以支持对异地分布对象的访问,其核心机制称为对象请求代理(Object Request Broker,ORB),从而屏蔽不同操作系统、不同语言环境的差别,将异构分布式系统"转化"为面向开发人员的一台虚拟计算机、单一的开发环境。DOO 不仅使 OO 的优点在异构分布式环境下得到保持,更重要的是大大简化了异构分布式软件开发工作的复杂性。

第五代(90 年代中期至今)软件技术是以 COM/DCOM、CORBA3.0、J2EE、.NET 等为代表的软件构件技术和中间件[10-12]。

软件构件是一种定义良好的独立、可复用的软件成分,包括功能模块、被封装的对象类、软件框架和软件系统模型等。一个软件可被切分成一些构件,这些构件可以单独开发、单独编译、单独测试与发布。当所有的构件开发完成后,把它们组合在一起就可以得到完整的应用系统。

COM/DCOM、CORBA3.0、EJB 等是构件技术在 Internet 中的延伸,其目标是提供基于 Internet 的系统平台异构性和互操作性,实现企业应用和数据资源的整合。这种构件在开发出来后被一次性部署到 Internet 中,所有应用只需要能够连入 Internet,就可以集成和使用其提供的服务。

由于不同的中间件标准又造成了事实上的异构性,催生了试图整合各种中间件技术的 Web Services[13]。Web Services 是一个建立可互操作的分布式应用程序的新平台,它将 Web 上的所有内容都封装为可访问的服务,通过一整套标准规范 Web 服务的描述方式、通信协议和调用机制,使大规模的资源共享成为可能。

2 促进软件形态发展的因素

从软件形态的演化历史来看,软件技术的发展与三个因素紧密相关[26]:

问题:应用需求的不断发展,对软件技术不断提出新的要求,成为技术进步的原动力,同时也是技术进步的最大受益者。

平台:平台决定了软件的生存形态和演化方向。从无操作系统的裸机,到带有操作系统的单机,到局域网再到 Internet,平台的不断发展产生了更多、更复杂的异构问题,促使软件技术持续更新。

方法:软件开发方法是在特定的平台上分析问题和获得解的指导方法。从以算法为中心,到以对象为中心,到以构件为中心,到以服务的集成为中心,软件开发方法同样经历了质的变化。

以上三个因素要求软件向上要具有适应应用需求的特性,向下要具有适应计算平台的特性,这些都是造成软件构造趋于复杂的直接原因。

3 软件形态的发展趋势——Internetware

20 世纪 90 年代以来,信息技术范围内最具有深远影响的技术进步是 Internet 的出现和普及,并迅速发展成为当今世界上覆盖面最广、规模最大、信息资源最丰富的计算机信息网络,深刻地改变了传统的技术以及应用、业务和产业模式。目前,计算机应用已经全面进入 Internet 时代[15]。Internet 作为一个技术复杂的网络平台,促进了新的底层技术、新的操作系统、新的编程语言、新的软件开发方法学和新的应用领域的出现。为更有效地进行 Internet 上的软件开发,充分利用 Internet 的潜力,需要屏蔽网络的复杂性和异构性,使我们能够跨越网络透明访问各种信息资源并协同处理[16],实现"The Network is Computer"的概念[17]。软件形态必须适应于这种变化趋势。

Internet 环境的特点在于[18][19][26]:

(1)基础平台的开放性、动态性和多变性;

(2)共享资源的多样性(信息、计算、服务);

(3)无统一控制的分布性;

(4)结点的高度自治性和不可预测性;

(5)结点链接方式的开放性和动态性;

(6)软硬件设施的多重异构性;

(7)网络连接环境的多样性;

(8)使用方式的个性化和灵活性。

以上特点使 Internet 成为一个巨型、高效、高信度、统一的虚拟资源平台,为用户提供了一个全方位的信息服务基础设施。

Internet 的特点使之适合于提供以下开发的应用模式[19]:

(1)连接信息:如 WWW 应用;

(2)连接企业:如电子商务;

(3)连接计算资源 如网格；

(4)连接服务 如 Web Service。

同时，这些特点决定了未来的软件系统需要能够在 Internet 环境下具备开放的结构，拥有动态协同、在线演化、环境感知和自主适应的能力；软件系统中的实体元素应被组织为分布、自治、异构的构件，要具有较高的抽象层次以利于复用，具有独立性、主动性和自适应性；实体间要能够互连、互通、协作和联盟，实现多种静态连接和动态合作。对于这种新的软件形态，北京大学的杨芙清院士、南京大学的吕建教授、中国科学院数学研究所金芝研究员都作过表述，将之称为“网构软件”(Internetware)[14][18][19]。

网构软件将在传统 Internet 应用模式的基础上，进一步深化开放环境下的资源共享，将信息、应用、计算资源、服务连接内容都视为网络资源，通过对资源的连接实现从信息 Web 到 Software Web 的跨越[19]。

4 网构软件的主要特征

网构软件是 Internet 开放、动态和多变环境下软件系统基本形态的一种抽象，既是传统软件结构的自然延伸，又具有区别于传统软件形态的独有的基本特征[18][19]，表述如下：

自主性：软件实体具有相对独立性、主动性和自适应性，是网络上的自治单元；

演化性：结构和实体的演化能力，体现为组成软件的元素数目可变性、元素间结构关系的可调节性和形态的动态可调性；

协同性：软件实体之间具有多种方式的互联、互通、协作和联盟；

多态性：软件实体具有目标制导和多目标的特征协同能力，从而使系统的效果体现出相容的多目标性；

反应性：软件实体具有感知外部运行和使用环境的能力。

网构软件的开发和演化过程与传统的功能分解和有序控制不同，强调预先开发并部署具有主动性的功能实体，然后通过查找符合需要的实体及实体的聚合形成软件系统，将需求变化映射为实体的重新选择和重新连接，从而促进软件系统对于网络变化特点的适应能力。

5 网构软件的研究领域和前景

在 Internet 平台下，传统软件工程基于实体驱动和确定目标、有序控制的开发模式开始让位于 Internet 下基于协同驱动和动态目标、实体聚合的开发模式；计算单元的开放性要求将自顶向下的需求获取和自底向上的需求支撑相结合，同时软件的动态演化性要求将自动需求建模和自主捕获需求变化相结合，实现软件工程的智能化。因此，智能软件工程成为网构软件的研究方向之一，包括用户主导的需求工程方法、面向本体的领域模型库、基于本体的需求自动获取、需求变化管理和需求模型演化、Internet 环境下的软件协同开发技术、基于知识的需求管理、软件工程知识管理等内容[20][21]。

网构软件所要求的实体分布性、自主性和协同性等极大地依赖于智能 Agent、移动 Agent 和多 Agent 系统等方向的研究，目前主要侧重在 Agent 的认知模型和理论、多 Agent 系统的体系结构、Agent 的协作与协商、Agent 计算环境的异构性与安全性、基于 Agent 的构件软件框架、面向 Agent 的软件方法学等[22][23]。国外经过近 20 多年的研究，Agent 理论与技术有了长足的发展，已经在很多领域中得到了应用，而国内起步相对较晚。

网构软件的在线演化、环境感知等能力 Internet 所有信息都具有计算机能够理解和处理的意义，对这些信息语义的表示和处理机制构成了语义网(Semantic Web)的研究领域[24]。关于语义网中的层次关系、知识表示、本体论、智能 Agent 以及在此之上构建的本体和逻辑推理规则等都是其重要的研究内容[25]。

通过对网构软件的研究，将能够更加充分地发挥 Internet 的计算潜力，在新的软件开发方法基础上推动新型软件形态的普及，促进软件生产率和质量的提高，深化软件的应用范围，带动系统软件、应用软件的发展，从而建立更加良性的软件产业链，研究前景非常广阔。(收稿日期：2003 年 2 月)

参考文献

1.Department of Defense Parametric Estimating Initiative Handbook[M].Second Edition,http://www.ispa-cost.org/PEIWeb/newbook.htm,Spring,1999

2.Jim Larus.The Gap Between Software Research and Practice[C].In:Talk on Dagstuhl 10th Anniversary Celebration,2000-08

3.E W Dijkstra.Go To Statement Considered Harmful[J].Communications of the ACM,1968;11(3):147~148

4.Pressman R Software Engineering:A Practitioner's Approach[M].Fifth Edition,McGraw-Hill,2001

5.Ole Johann Dahl,Nygaard Kristen.Simula--An Algol based simulation language[J].Transactions of the ACM,1966;9(9):671~678

6.Jacobson I.Object-Oriented Software Engineering,A Use Case Driven Approach[M].Addison-Wesley Publishing Company,1992

7.Object Management Group et al.The Common Object Request Broker:Architecture and Specification[S].Version 2.4.2,2001

8.Yen-Ping Shan,Ralph H Earle.Enterprise Computing with Objects:From Client/Server Environments to the Internet[M].Addison Wesley Longman,1998

9.王怀民.国防科技大学技术报告:分布对象技术[R].1999

10.OMG.CORBA 3.0.2 Specification[S].http://www.omg.org/technology/documents/formal/corba_iiop.htm,2002

11.Sun Microsystems.J2EE 1.4 Platform Specification[S].2003

12.Microsoft Corporation.Microsoft COM homepage.http://www.microsoft.com/com,2000

13.Microsoft Corporation.Global XML Web Services Architecture[R].Technical Report,2002

14.吕建.南京大学计算机软件新技术国家重点实验室学术报告:面向对象技术研究与发展[R].2003

15.Ian Foster,Carl Kesselman et al.The Grid:Blueprint for a New Computing Infrastructure[M].Morgan kaufmann Publishers Inc.San Francisco,California,1997

16.冯玉琳，黄涛，金蓓弘.网络分布计算和软件工程[M].科学出版社，2003

17.Steven D Gribble,Andrew Whitaker,Marianne Shaw.Lightweight Virtual Machines for Distributed and Networked Systems,Talks in HP Labs and Sprint Labs,2001-07-07

18.杨芙清.2003 全国软件与应用学术会议(NASAC)大会报告:软件技术、软件产业与人才培养[R].2003-11-14

(下转 59 页)

失和 $LossSum=\sum_{s_l\in D} Loss_l$ 最小的实例进入 D，逐步减少测试集 T 实例数，直到 T 为空。

4.3 算法描述

输入：训练集 $D=\{s_1,s_2,\cdots,s_\eta\}$，$s_i=<x_i,c_i^0>$，其中 c_i^0 是 x_i 的已知类别；测试集 $T=\{x'_1,x'_2,\cdots,x'_\mu\}$，易知在第 2 节约定前提下有 $\eta+\mu=n=|S|$。

输出：分类器 C。

算法实现：

(1)在训练集 D 的基础上学习分类器 C，求出 D 中所有实例 s_l 的类条件概率 ρ_l。

(2)如果 $T=\varnothing$，算法结束，返回分类器 C，否则继续。

(3)令 $LeastLoss$=某个足够大的数；对 T 中每个实例 x'_p，重复以下过程：

(3.1)利用公式(3)，学习获得 x'_p 的类别标签 c'_p；

(3.2)在 $D'=D+\{<x'_p,c'_p>\}$ 的基础上，重新计算 D 中所有实例 s_l 的类条件概率 ρ'_l 及其估计损失 $Loss_i$，从而得到估计损失和：$LossSum=\sum_{s_i\in D} Loss_i$；

(3.3)如果 $LeastLoss>LossSum$，则令 $LeastLoss=LossSum$，$x_{selected}=x'_p$，$c_{selected}=c'_p$，并暂存 D 中所有训练实例 s_l 在(3.2)中计算的 ρ'_l。

(4)更新分类器参数：$D=D+\{<x_{selected},c_{selected}>\}$，$T=T-\{<x_{selected}>\}$，把(3.3)中暂存的所有原训练实例 s_l 的 ρ'_l 赋给相应的 ρ_l，在新的训练集中计算并保存新加入的 $s_{selected}=<x_{selected},c_{selected}>$ 的类条件概率 $\rho_{selected}$，转向(2)。

5 算法原理分析

在本增量学习算法中，需要为每个训练实例 s_l 保存以前的类条件概率，用以与后来的概率值结合计算总估计损失，这是该算法额外的空间需求。在时间性能上，为了提高精度，在选择每个测试实例时，都要重新计算训练实例的类条件概率和估计损失，其计算训练实例估计损失的基本操作共进行 $\sum_{i=1}^{|T|} i(|S|-i)$ 次，时间开销远远小于 $N(|S|^3)$，因此该算法是完全可行的。

很容易就可以分析出该算法的学习功能：

首先，算法充分地结合了先验知识，这里就是在结合先验知识的基础上来计算估计损失的。

其次，训练实例估计损失权重系数 $\lambda_l=\rho_l\exp(\Delta\rho_l)$ 恰如其分地反应了新实例对训练实例的影响作用，对于 D 中实例 s_l 而言，ρ_l 越大，说明该实例比较充分地反映了实例集的整体特点，其对外来实例所产生影响的敏感程度应该相对较高，所以把 ρ_l 作为 λ_l 的一个因子。另外，ρ'_l 与 ρ_l 之间的关系必是下述三种情况之一：

(1)$\rho'_l>\rho_l$，此时 $\Delta\rho_l>0$，说明新实例的加入强化了实例 s_l 的表达力，$|\Delta\rho_l|$ 越大，s_l 对新实例的敏感程度应该越小。

(2)$\rho'_l=\rho_l$，即 $\Delta\rho_l=0$，说明此时新实例对 s_l 的表达力没什么额外影响。

(3)$\rho'_l<\rho_l$，此时 $\Delta\rho_l<0$，说明新实例弱化了实例 s_l 的表达力，$|\Delta\rho_l|$ 越大，s_l 对新实例的敏感程度就越大。

为了反映这一事实，在 λ_l 定义式引入了因子 $\exp(\Delta\rho_l)$。由上述分析可知，引入估计损失权重系数以后，在综合各种因素的基础上，该算法保证了相对可靠的测试实例优先选择，从而优化分类器性能，提高分类精度。

最后，贝叶斯学习方法建立在属性间条件独立假设前提下。属性独立性假设往往不能满足，上述算法中反复用训练实例检验的过程弱化了属性间相关性的负面影响。

6 结束语

贝叶斯方法的统计特性和处理先验知识的能力使得贝叶斯分类器有明显的性能优势，自上世纪 80 年代以来越来越得到研究人员的重视。该文针对增量分类器学习过程提出了一种改进的综合考虑类别估计损失的学习算法，在增量学习过程中，每次选择合适的测试实例时，用当前训练集的实例进行损失检验，努力让与当前训练集拟合度较好的测试实例被优先学习。这种算法有效地结合了先验知识并在一定程度上解决了先验信息的传递的问题。这一思路应该能延伸到其它学习过程中，如贝叶斯网络结构学习，参数学习，神经网络等，这也将是下一步研究的重点。(收稿日期：2003 年 9 月)

参考文献

1.Nir Friedman，Dan Geiger.Bayesian network classifier[J].Machine Learning，1997，29：131~163

2.宫秀军，刘少辉，史忠植.一种增量贝叶斯分类模型[J].计算机学报，2002，25(6)：645~650

3.Dominigos P，Pazzani M.On the optimality of the simple Bayesian classifier under zero-one loss[J].Machine Learning，1997，29：103~130

4.Langley P，Sage S.Induction of selective Bayesian classifiers[C].In：Proc 10th Conference on Uncertainty in Artificial Intelligence.Seattle，WA：Morgen Kaufmann，1994：399~406

5.林士敏，田凤占，陆玉昌.用于数据采掘的贝叶斯分类器研究[J].计算机科学，2000，27(10)：73~76

(上接 40 页)

19.金芝.中国科学院管理、决策与信息系统重点实验室学术报告：网构软件与基于知识的需求工程[R].2003

20.杨叶，李明树.用户主导的面向领域的需求分析方法[J].计算机工程与设计，2000，21(2)：21~25

21.李明树，卢梅，赵欣培.User-Driven Domain-Specific Software Requirements Analysis[C].In：13th Int'l Conf on Syst Engineering(ICSE'99)，Nevada，USA，1999-08：35~40

22.吕建，张鸣，廖宇等.基于移动 Agent 技术的构件软件框架研究[J].软件学报，2000，11(8)：1018~1023

23.N R Jennings，M J Wooldrige et al.Agent Technology[M].Springer，1998

24.T Berners-Lee，J Hendler，O Lassila.The Semantic Web[J].Scientific American，2001，284(5)：34~43

25.J Hendler.Agents and the Semantic Web[J].IEEE Intelligent Systems，2001，16(2)：30~37

26.陶先平，吕建.南京大学计算机软件研究所学术报告：基于 Internet 的软件 Agent 技术[R].2003