

文章编号:1001-9081(2006)04-0847-03

软件分布式开发中协同工作支持框架的设计与实现

刘景华, 刘 强

(清华大学 软件学院, 北京 100084)

(jh-liu03@mails.tsinghua.edu.cn)

摘要: 为了有效地支持软件分布式开发, 提高软件协同开发环境的易用性和有效性, 从协同理论和应用工具两方面入手, 在集成了多种协同应用软件的同时, 融入了对参与者的形式描述, 提高了开发环境的智能性和可定制性, 进而提高了协同开发的质量和效率。

关键词: 软件工程; 分布式开发; 协同工作; 形式描述

中图分类号: TP311.5 **文献标识码:** A

Design and implementation of supporting framework for collaborative work in distributed software development

LIU Jing-hua, LIU Qiang

(School of Software, Tsinghua University, Beijing 100084, China)

Abstract: In order to effectively support distributed development and make software collaborative development environment more effective and comfortable to use, collaborative theory and application tool were integrated with formal description. This practice effectively improves the intelligence and scalability of the environment, along with the quality of software and the productivity of the development.

Key words: software engineering; distributed development; collaborative work; formal description

0 引言

随着软件规模和复杂度的不断增长, 软件开发越来越体现出多角色参与, 多环节配合的工程项目的特点, 涉及到大量的协同工作。尤其在计算机网络迅速普及, 跨国软件公司数量迅速增长的背景下, 软件外包及组件化分布式开发成为许多公司降低成本、提高效益的有效策略^[8,9,12]。此外, 分布式协同开发还有助于促进不同背景和专长的开发人员间各种能力的互补。但是, 分布式开发也给传统的软件工程理论和方法提出了新的课题^[7-9]: 远程交流需要新的手段以保证沟通的及时性和频率; 不同地点的开发团队间如何建立信任, 如何协调进度, 如何解决矛盾; 如何消除软件项目各参与方的信息鸿沟等。

软件工程中参与者间的沟通和协调始终是工作的核心难点之一^[1,2], 这一难点在分布式开发中被放大了。为了解决沟通和协同工作的困难, 研究人员综合分析了相关理论, 或是从实践经验出发总结了沟通的有效方式和协同支持工具^[4-6,10]。本文综合了以往理论和实践中的研究成果, 应用计算机支持的协作工作 (Computer Supported Cooperative Work, CSCW) 和计算机辅助软件工程 (Computer-Aided Software Engineering, CASE) 理论与工具, 并融入对软件开发各个利益干系人的形式描述, 构建了一个基于网络的软件协同开发环境 (Software Collaborative Development Environment, SCDE), 力图提高软件分布式开发的效率和质量。

1 软件开发协同工作的分析

1.1 协同工作与沟通方式

软件协同开发中的关键点和难点之一, 就是参与者间的

沟通和协调, 即开发人员、客户和资源提供商间的协同工作, 其中开发人员间的协同工作占最大比重。为此, 首先以开发人员为例, 对他们在软件开发环境中的工作进行分析研究。对开发人员的工作进行了较长时间的跟踪分析后, 归纳出三种工作类型: 预先计划的协同工作, 如定期的会议、报告等; 随意性的协同工作, 如成员间随意的交流等; 个人工作, 如独立完成分配给个人的任务等。

其中, 前两个是协同工作的部分, 是我们研究的重点。开发人员在上述工作类型中占用的时间比例如图 1 所示, 随意性的协同工作占据了大部分工作时间, 体现了该类型协同工作的重要性, 因此我们设计的软件协同开发平台有针对性地支持了上述各个工作类型, 尤其注重对随意性沟通的需求。

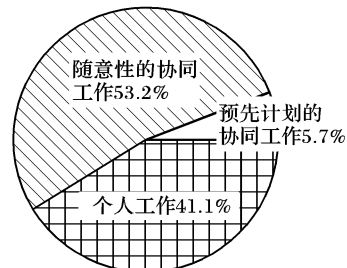


图 1 开发人员协同工作时间比例

在分布式环境下, 软件开发参与者间的沟通方式可以根据介质的不同分为: 基于传统通信的手段, 如纸质信件, 电话等; 基于网络的通信手段, 如电子邮件、网络白板、IP 电话和网络即时通讯软件等; 面对面交谈。

也可以根据沟通的信号表示形式不同分为: 文本形式, 语音形式和视频形式。

收稿日期: 2005-10-12; 修订日期: 2005-12-03

作者简介: 刘景华 (1973-), 男, 福建福州人, 硕士研究生, 主要研究方向: 软件工程、软件项目管理; 刘强 (1963-), 女, 北京人, 副教授, 硕士, 主要研究方向: 软件工程、软件项目管理。

经研究发现,开发参与者更倾向于用电子邮件来进行文本形式的通信,用电话进行语音形式的通信。但是,随着应用的普及,参与者已经逐渐适应使用网络即时通讯软件支持语音和文本通信。

1.2 软件开发参与者的形式描述

软件协同开发环境必须有效地支持软件开发各类参与者的协同工作,这就需要对各类参与者的属性和从事的工作有系统的了解和规划。但是,作为个体和团体的人的复杂性,使得我们很难全面而系统地在软件工程中构建处置人为因素的完备而准确的框架。受文献[2]的启发,通过对软件开发参与者的属性和行为进行系统的分析,提取了若干要素,构造了形式描述,并尝试通过这种技术手段系统地支持和管理软件开发过程中参与者的协同工作,同时为构建高效率的协同开发环境提供底层运行机制。

从项目管理的角度,可以将软件开发的所有参与者称作软件项目利益干系人。参与软件开发的个体,包括软件架构师、项目经理、设计人员、编码人员、测试人员和维护人员等,他们属于软件开发团体;也包括软件外包供应商代表,软件开发所需软硬件平台提供商代表,他们属于资源供应商团体;还包括软件订单客户、软件最终用户等,他们属于软件客户团体。上述个人及团体都属于软件项目的利益干系人,对他们的形式描述必须体现彼此在软件开发项目中的关联和影响。我们对于个体的描述如下:

$$I = (C, R, K, J, Pc, Sr, Pr) \quad (1)$$

其中:

I (Individual): 个体的名称; C (Community): 个体所在的团体名称; R (Role): 个体在团体中扮演的角色,代表了个体所拥有的权限和能够实施的工作; K (Knowledge): 个体所拥有的知识,包括了编程语言,工作技能和特长等; J (Job): 个体所从事的工作; Pc (Personal characteristics): 个体的性格特点,可能影响其在工作中的表现和决策,不同的性格特点适合不同类型的工作; Sr (Superior): 个体的上级; Pr (Partner): 个体的合作伙伴,是个体最经常沟通的对象。

对应的团体描述如下:

$$C = (Ms, Rs, L, W, Cp, Lr, Re) \quad (2)$$

其中:

C (Community): 团体的名称; Ms (Members): 团体的成员汇总; Rs (Roles): 团体成员的角色汇总; L (Language): 团体内部的沟通语言,这种语言是广义的语言,包括了口头、书面和肢体等语言,也包括了记录符号、事物标记等语言形式; W (Work): 团体执行的工作,它通过 WBS (Work Breakdown Structures) 方法获得的分层次的工作内容分解结构,细化到由团体中的个体能够执行的工作单元; Cp (Choice principles): 团体的选择原则,是团体进行决策的依据; Lr (Leader): 团体的领导者; Re (Relation): 团体与其他团体的关系。

2 软件协同开发环境的架构

基于上述分析和研究,我们综合了多个领域的知识,构建基于网络的协同工作平台,其主要的支撑理论和技术如图 2 所示。

该软件协同开发环境(SCDE)结合了软件开发环境(Software Development Environment, SDE)和群件(Groupware)^[11]的概念,是针对分布式开发的特点而设计的。许多已有的软件开发环境产品侧重于提供业务及沟通的工具支持,SCDE 不仅提供相

关工具,还内置了对软件开发参与者的形式描述,并据此自动配置用户的工作环境,提供个性化平台支持,提高工作效率。

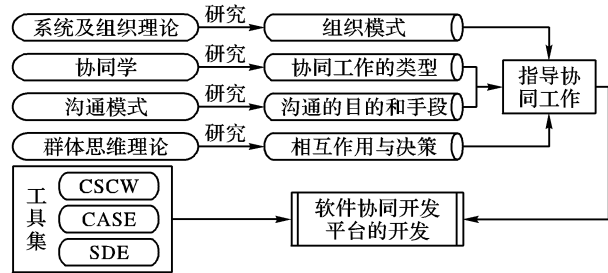


图2 软件协同开发环境支撑理论及技术

支持软件协同开发的环境有多种架构,有按功能模块实现的架构^[10],有根据支撑技术实现的架构^[6],还有用 P2P 网络模型实现的架构^[3]。我们的软件协同开发环境的架构如图 3 所示。这是一个细化了的三层架构,其中应用程序部分包括了通用应用程序和协同支持程序,数据库和应用程序均增强了安全保证。针对用户对随意性协同工作的需求,重点增强了即时通讯软件的功能,提供了多媒体网络白板,即时文本和语音通讯。

在这个架构中,业务过程逻辑、形式描述和工作流引擎构成了系统运行的驱动机制。其中,形式描述对于动态组织管理、任务管理和用户工作环境定制发挥了重要作用。对于形式描述的实现方法和应用机制如图 3 所示。

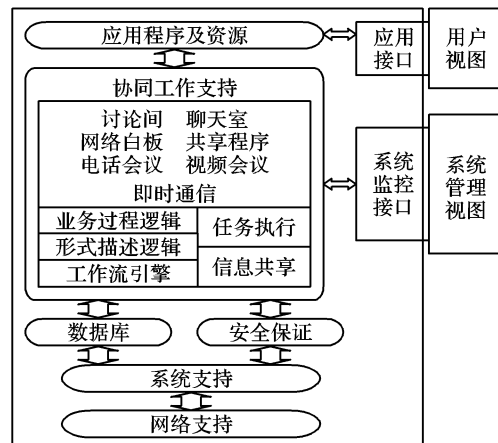


图3 软件协同开发环境架构

1) 以 XML 对用户个人及团体进行形式描述,XML 中的元素名代表形式描述中的参数项,元素值代表对应的参数值。在 SCDE 的系统管理视图中提供编辑形式描述 XML 文件的界面和工具。

2) 在 SCDE 中内置的 XML 解析机完成对形式描述的元素名、属性和元素值的提取。在 SCDE 软件的动态组织管理、任务管理和用户工作环境定制模块中调用 XML 解析机提取形式描述中的相关参数。

形式描述的优点就突出体现在定制的简易性和对动态管理的支持。以用户工作环境的定制为例,该工作环境主要由三个部分构成:

(1) 个人任务执行区,用于接收分配的任务,提交完成的任务,制定个人工作日志等。该部分的任务提取主要依据公式(1)中的属性 J 和公式(2)中的属性 W 生成。

(2) 沟通交流区,用于与其他成员进行沟通。该部分的沟通对象主要依据公式(1)中的属性 Sr、Pr 和公式(2)中的属性 Re 生成。

(3)信息共享区,用于发布和浏览共享的信息。信息的访问权限主要依据公式(1)中的属性C和R确定。

如果一个开发人员更换了开发组,其所要执行的任务,工作沟通对象和信息访问权限都要发生变化。如果没有工作环境的自动生成机制,用户就需要自己重新定位工作任务,设置沟通对象,搜索信息,不但繁琐,而且容易出现遗漏和差错;而通过形式描述,只要对形式描述中相应的参数作简单的修改,系统就能够根据这些参数,在用户下一次登录时自动生成工作环境。

3 应用情况与统计数据

在采用新架构开发 SCDE 之前所使用的一套 SDE 软件主要是用于支持软件开发的技术流程,没有采用形式描述逻辑和工作流引擎,用户需要自行定位需要的信息和沟通对象;提供的沟通手段也仅局限于电子邮件和网络会议功能;而且系统不向“资源供应商”和“软件客户”开放。

采用新的协同开发环境后,我们实施了三个软件开发项目,系统监控程序跟踪记录了用户使用情况,将这些统计数据与老系统运行时记录的两个项目的统计数据进行了对比分析,发现在几个方面获得了有效的改进。主要表现在:开发人员的工作效率得到了提高;软件开发的周期缩短了,返工的比例显著降低;客户参与度提高。

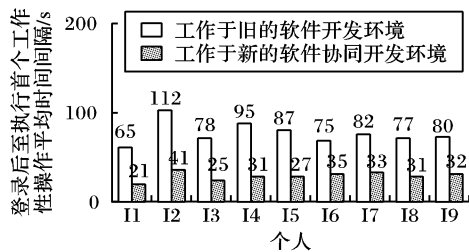


图4 登录后至执行首个工作性操作平均时间间隔

以下列举几个主要统计数据以体现相关方面的改进。图4显示了在30个工作日内9个用户登录系统后至执行首个工作性操作的时间间隔的平均值。考虑到用户硬件的差异,只针对单个用户进行纵向比较。可以看到,在新系统中,9个用户的该项数据的值都有了显著减少,从53.3%至69.0%,平均减少了63.1%。其主要原因是新系统为用户自动生成工作环境,减少了不必要的手工操作。

另一项反映软件开发效率的数据是项目建立完整需求的时间。如图5所示,新系统中的项目P3、P4和P5建立完整需求的时间,比旧系统中的项目P1和P2有了显著减少。其主要原因是新系统中客户团体有效地参与了开发过程,减少了获取需求的中间环节。

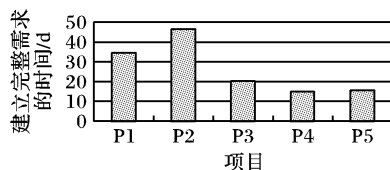


图5 建立完整需求的时间

体现软件开发质量和效率的另一项典型数据是产品缺陷从发现到修正的时间间隔。如图6所示,10个代码容量相近的软件包,其中B1至B5是旧系统上开发的软件包,其均值是58d, B6至B10是新系统上开发的软件包,其均值是18.8d。产品缺陷从发现到修正的时间间隔显著减少,其主要原因是新系统提高了缺陷报告和处理反馈的及时性。

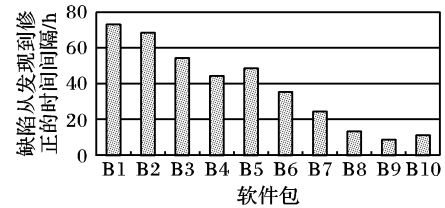


图6 产品缺陷从发现到修正的时间间隔

4 结语

对于分布式环境下的软件协同开发的支持,不仅仅需要恰当的工具,还需要对软件开发参与者的应用倾向和角色特点进行系统的分析,根据不同个体和团体的属性定制个性化的工作环境,明确访问权限,保持沟通的积极性和有效性。在新的软件协同开发环境的实践表明,从这两个方面同时着手,能够有效提高软件协同开发的质量和效率。

下一步将研究如何有效地进行任务划分和程序模块分解,使之更适合分布式环境下的软件开发,降低协同工作的复杂度。

参考文献:

- [1] FREDERICK P, BROOKS JR. The Mythical Man-Month[M]. 2nd Edition. New York: Addison Wesley Longman Inc, 1995. 13-26.
- [2] JEFFREY HJ. Addressing the Essential Difficulties of Software Engineering[J]. Journal of Systems and Software, 1996, 32(2): 157-179.
- [3] BOWEN S, MAURER F. Designing a Distributed Software Development Support System Using a Peer-to-Peer Architecture[A]. Proceedings of the 26th Annual International Computer Software and Applications conference[C]. 2002.
- [4] ROBILLARD PN, ROBILLARD MP. Types of collaborative work in software engineering[J]. The Journal of Systems and Software, 2000, 53(3): 219-224.
- [5] MCCHESNEY IR, GALLAGHER S. Communication and co-ordination practices in software engineering projects[J]. Information and Software Technology, 2004, 46(7): 473-489.
- [6] WU L, SAHRAOUI H, VALTCHEV P. Caribou: a Supporting Environment for Software e-Development[A]. Proceedings of The 2005 IEEE International Conference On e-Technology, e-Commerce and e-Service[C]. 2005. 736-739.
- [7] MARTTIIN P, LEHTO JA, NYMAN G. Understanding and Evaluating Collaborative Work in Multi-Site Software Projects—A Framework Proposal and Preliminary Results[A]. Proceedings of the 35th Hawaii International Conference on System Sciences[C]. 2002.
- [8] HERBSLEB JD, MOITRA D. Global Software Development[J]. IEEE Software, 2001, 18(2): 16-20.
- [9] MOCKUS A, HERBSLEB J. Challenges of Global Software Development[A]. Proceedings of the 7th International Software Metrics Symposium[C]. 2001. 182-184.
- [10] SAEKI M. Communication, Collaboration and Cooperation in Software Development—How Should We Support Group Work in Software Development[A]. Proceedings of Asia-Pacific Software Engineering Conference[C]. Brisbane, Australia, 1995.
- [11] DEFRANCO-TOMMARELLO J, DEEK FP. Collaborative Software Development: A Discussion of Problem Solving Models and Groupware Technologies[A]. Proceedings of the 35th Hawaii International Conference on System Sciences[C]. 2002.
- [12] LAYZELL P, BRERETON OP, FRENCH A. Supporting Collaboration in Distributed Software Engineering Teams[A]. Proceedings of the 7th Asia-Pacific Software Engineering Conference[C]. 2000.