

基于群集智能的软件开发过程改进

张亚琦¹ 张英朝¹ 李 炳²

(北京系统工程研究所 北京 100101)¹ (中国人民银行 北京 100800)²

摘要 复杂软件系统正在成为一种泛在的新型软件形态,在社会、经济、军事活动中占据越来越重要的地位。然而应用传统的软件工程方法进行复杂软件系统开发十分困难。首先分析了复杂软件系统开发过程的特征,发现了其中蕴含的生物智能和“共进化”机制;进而提出一种基于群集智能的复杂软件系统演化模型;最后,基于该模型给出了复杂软件系统开发过程改进的方法建议。

关键词 复杂软件系统,群集智能,演化,开发过程改进

中图分类号 TP311.52 文献标识码 A

Process Improving in Software Development Based on Swam Intelligence

ZHANG Ya-qi¹ ZHANG Ying-chao¹ LI Bing²

(Beijing Institute of System Engineering, Beijing 100101, China)¹ (People's Bank of China, Beijing 100800, China)²

Abstract Complex software system is now becoming a new ubiquitous form of software, playing a more and more important role in social, economic and military activities. However, the traditional software engineering methods encounter difficulties when applied to complex software developing. Firstly, we analyzed the characteristics of complex software development process, from which we found implicated biology intelligences and a co-evolution mechanism. And then we raised up an evolution model based on swam intelligence. At last, based on the model, we gave measures to improve the process of complex software system developing.

Keywords Complex software system, Swam intelligence, Evolution, Process improving

1 引言

近年来,随着网络在各业务领域的深度融合发展,万物互联带来了大量数据与信息,软件复杂程度持续增长,许多大型软件在内部结构、外部交互、演化方式等方面表现出与传统分布计算软件迥异的特征,具有了明显的复杂系统特征:1)由多数量的模块组成,模块间互联互通关系复杂;2)模块本身具有自演化能力;3)系统是动态的、不停止的、不可预测和非线性的;4)系统具有一定的适应性和反馈能力^[1]。

文献[2]给出复杂软件系统定义:复杂软件系统是指由大量局部自治软件系统持续集成、相互耦合关联而成的大型软件系统,此类软件系统与其所作用的社会系统和物理系统密切相关,系统要素之间的耦合交互关系动态变化且日趋复杂,整个系统的行为难以通过各自软件系统特征的简单叠加以刻画,表现出“成员异质、边界开放、行为涌现、持续演化”等一系列新的性质。

近年来越来越多的软件项目失败,最主要的原因是软件系统的复杂性越来越高^[3]。软件开发过程管理是组织不断改进软件开发能力和提高软件质量的第一要素^[4],对软件项目的成败负有重要责任。传统基于还原论的软件开发方法受到大规模工业生产线模式的影响,旨在以精确和严密的计划产出符合事先需求约定的软件产品,这一方法用在复杂软件系统开发时显现出局限性,导致软件项目失败。

钱学森提出,凡是不能用还原论方法处理的或不宜用还原论方法处理的问题,要用或宜用新的科学方法处理的问题,都是复杂性问题^[5]。采用仿生途径研究系统复杂性是一种重要方式和一个恰当的选择。群集智能(Swarm Intelligence)能够反映一般复杂系统的主要特征,包括涌现引起的复杂性(行为特性)、适应造就的复杂性(进化特性)和组合出现的复杂性(结构特性),其受到复杂系统研究者越来越多的关注^[5]。

本文分析了复杂软件系统开发过程特征,尝试将群集智能应用于软件工程领域,提出一种能够适应复杂软件系统生长和代谢特征的“共进化”模型,基于该模型给出复杂软件系统开发过程改进方法,期待赋予复杂软件系统持续演化与发展的生命力。

2 复杂软件系统开发过程特征

复杂软件系统除具有一般性的系统特征如多元性、相关性和整体性以外,还具有一些更为重要的特征。

2.1 分布式

复杂软件系统是一个分布式系统,多基于广域网实现,由高度异构、相互独立的自治系统持续集成而成,共同完成某项任务,整体依赖于局部系统功能但又不单独依赖于某一局部系统。局部系统具有较强的自治性且持续演化,长期自组织行为已使其周遭的社会和物理系统形成固有行为模式和反馈-响应环境,不同自治系统在体系结构、数据模型、软硬件平

张亚琦(1981—),女,硕士,助理研究员,主要研究方向为软件工程, E-mail: Wingwinglili@foxmail.com; 张英朝(1977—),男,博士,副研究员,主要研究方向为体系工程; 李 炳(1981—),男,博士,工程师,主要研究方向为互联网金融。

台、应用环境等方面存在差异,所涉及的人和管理域还可能存在利益冲突^[2]。

复杂软件系统开发过程既要兼顾各局部系统利益,确保系统可用性,又要消除系统间的冲突与重合,确保系统可集成性,还要考虑系统整体能力和技术发展水平,确保系统的先进性与延续性。

2.2 自组织自学习

复杂软件系统由于存在结构复杂性、过程复杂性等特点,开发过程很难一次完成,往往需要多次迭代反复,最终实现持续稳定的运用与发展。复杂软件系统在设计初期很难提出全面、具化、准确的需求,初期提出的需求在后续开发过程中可能发生变化或调整,技术革新也会催生出新的需求。系统需要在开发过程中不断从内部和外部学习,获取、选择、剔除和确认需求,并选择最优方案来实现需求。

复杂软件系统的开发过程应是自组织自学习的过程,最终使得系统发展由无序走向有序,由低级走向高级有序。

2.3 整体涌现性

复杂软件系统开发过程具有整体涌现性(Whole Emergence),当外界处于某种特定条件下时,局部系统微观聚集(有些是成员之间相互作用的结果,有些是物理和社会环境反馈产生)突然产生系统整体不可预见的结果,如“整体大于部分之和”或“整体小于部分之和”。这种涌现可以是功能涌现,也可以是结构涌现或模式涌现等^[5]。

复杂软件系统生命周期是非线性的,甚至可能是多维的。

2.4 正反馈

反馈是系统输出对输入的反作用,分为正反馈与负反馈,正反馈具有增补作用,能够促进事物发展。软件演化是指软件进行变化以具有新功能的过程^[6],是复杂软件系统对外界正反馈的体现,通常需要在保持系统持续运行、常态服务的基础上来实现动态适应,人类社会和物理世界存在固有的变化性将映射到复杂软件系统的技术维度上。

复杂软件系统开发过程强调正反馈的优化作用,更加强调适应变化、持续成长。

3 复杂软件系统的共进化机制分析

通过生物智能探索复杂系统已成为一种发展方向,复杂系统的自组织、自学习和涌现等特征与生物进化机制不谋而合。生物共进化是指紧密相关的种群之间的补充进化,也包括生物进化与环境进化之间的相互影响。复杂系统的共进化是指局部系统之间、系统与环境之间相互作用而演化。遗传算法模仿了生物遗传学和自然选择机理,采纳了选择、交叉、变异、迁移等自然进化模型,成为复杂系统建模与仿真的重要方法,基于遗传算法可将复杂软件系统的共进化过程分为3类。

(1)竞争。软件功能提供者之间存在竞争,任何一方的进步都会威胁另一方的生存能力,竞争者因为生存压力相互刺激对方而共进化。

(2)合作。复杂软件系统由多个局部系统分而治之和分工协作,局部系统承担某一专业工作,专业化程度越来越高以致其他系统不可替代,多个局部系统形成一个恰到好处的系统规模,并逐步向前进化。

(3)与环境协同。复杂软件系统的规模、计算能力和应用范围都严格受到环境约束。复杂软件系统要发展就要消耗环

境的供给,一旦这种消耗超过环境承受能力,环境无法自修复,就会给系统负反馈信号,系统作出相应调整以促进环境好转,而一旦环境好转,系统就继续发展。

复杂软件系统正是通过局部系统的竞争、合作、调整来实现整体共进化,完成更复杂的任务或比单一系统更高效的任务。

4 基于群集智能的复杂软件系统演化模型

群集智能是指简单主体通过交互作用所表现出来的不可预见的宏观职能行为的特性^[7]。它具有分布式结构特性,同样具有自组织、自学习和正反馈的特性。

将群集智能应用于软件工程领域,在软件开发过程中实施群体协同开发(种群合作)、开放资源共享(种群竞争)、运行状态监控和管理(种群与环境协同),来实现复杂软件系统演化(共进化)。

基于群集智能的复杂软件演化模型如图1所示,包括开发、分享和应用3个环节^[8]。3个环节的行为主体各不相同,各个环节内部也可能存在多个具有合作或是竞争关系的行为主体。3个环节如齿轮般各自运转、相互咬合,推动系统整体演进。

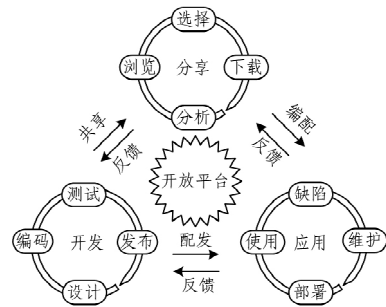


图1 复杂软件系统演化模型

开发环节包括设计、编码、测试与发布4个阶段,当接收到来自分享与应用环节的需求反馈时,开发团队启动开发环节生命周期,开发者或是合作、或是竞争,共同完成相关需求,定期将版本发布给分享与应用环节。

分享环节包括分析、浏览、选择与下载4类行为,将开发环节产生的各类产品、文档、资源共享给项目组甚至是更广阔的网络环境群体,使软件的特性和问题尽可能早的显露出来,同时攫取需求和技术改进灵感,加速软件产品传播、更新与积累过程,促进软件质量改进。

应用环节包括部署、使用、发现缺陷、维护4类行为,将从开发和分享环节获得的软件部署在运用场景中,通过实时监控、人工记录等方法收集系统缺陷与不足,反馈给开发团队或是共享发布方进行改进,从而促进系统内部及系统与环境之间的协调,为系统演化带来动力。

基于群集智能的复杂软件系统演化模型与传统软件生命周期模型相比,主要有以下不同。

(1)不再依靠单一团队或组织,充分发掘群体的智慧和创造力,引入未知的群体力量,创造协同创新的开发环境。

(2)不再封闭与垄断开发,引入共享与竞争机制,将软件产品按照规则共享给更广阔的群体,开放系统开发边界,刺激软件高质、高效演化。

(3)引入持续反馈机制,有效获取软件开发、共享、运行和环境反馈数据,通过对这些数据分析和挖掘,对软件系统开发

过程动态调整,使系统不断优化。

(4)系统生命周期不再有明显的线性阶段性,软件产品也不再是在生命周期结束时交付,而是软件持续改进、在线演化,系统能力提升以涌现的方式发生。

与传统软件开发模型相比,基于群集智能的软件系统演化模型具有先进组织模式和隐式协调控制机制,能够克服传统开发模式缺乏灵活性、组织单一性和学习能力低下的问题,实现自组织涌现的演化模式。

5 复杂软件系统开发过程改进

传统软件工程方法用于复杂软件系统开发已有许多不适应性,主要由于基于还原论的软件生命周期模型不适用于复杂软件系统。基于群集智能的复杂系统演化模型为复杂软件系统开发提供了新的思路与视角,可从以下方面改进软件开发过程。

5.1 建立软件开发、共享与演化平台

5.1.1 平台主要功能

软件开发、共享与演化平台是群集智能演化模型得以实现的核心条件,物理上可能由一个或多个小平台的组合而成。它为复杂软件系统开发提供协同开发、资源共享、软件集成、运行监控和在线演化等服务。

(1)协同开发服务

通过社交网络实现群体协同开发,对开发行为进行统计和分析,有序组织和快速定制软件生产。主要包括协同开发管理工具、协同开发分析服务和软件生产线框架。

1)协同开发管理工具:将项目论坛、协同编辑、邮件列表、即时通讯等交流分享工具和项目管理、配置管理、缺陷管理、持续集成等工业化生产工具集成在统一的开发环境中^[9],为群体化开发提供必要的沟通交流、感知互动和项目管理手段,实现软件创作和生产活动的深度融合。

2)协同开发分析服务:建立海量软件开发数据的获取和存储平台,集成多种开发活动度量工具,能够对开发者的能力水平、成长轨迹和贡献度进行精确分析^[10],这些数据也可以作为预估和评判软件质量的依据^[11]。

3)软件生产线框架:按照既定开发步骤,为软件开发者提供符合技术体制要求的开发环境、工具和组件服务,形成相应的软件生产线。

(2)资源共享服务

支持用户发布、检索、评估软件资源,将传统封闭、静态的软件构件存储模式转变为网络条件下开放、动态的软件资源分享模式,实现基于网络搜索的开放式软件资源获取与分享。通过建立软件价值评估模型,实现软件资源价值反馈闭环。主要包括软件资源管理和价值评估服务。

1)软件资源管理:建立软件资源统一封装、标识、版本发布和分类机制,支持软件资源特征分析,实现大规模软件资源的检索服务,提升软件资源发现与使用效率。

2)软件资源价值评估:根据用户对软件资源各种属性的不同关注点,对软件分享环节的反馈信息进行自动或者人工分析和评估,利用获取的反馈信息以及用户定义的需求模型对软件资源进行评估^[12],最终形成对软件资源关于某类属性(或整体价值)的等级评定。

(3)软件集成服务

通过对软件资源按照规范格式进行封装与验证,并组合在

统一的界面框架下部署使用,实现面向用户业务需求的软件资源集成运用。主要包括软件封装验证工具和界面集成框架。

1)软件封装与验证工具:支持将各类应用、服务、开发包、插件等按照标准格式封装成“包”(Package),并对其合法性和安全性进行验证。

2)界面集成框架:能够将各种“包”集成到具有统一风格的框架中,支持被集成“包”、框架和用户之间的消息传递。

(4)运行监控服务

通过植入监控探针获取软件运行状态信息,完成软件应用环节的反馈闭环。主要包括监控信息采集、监控信息分析诊断服务。

1)监控信息采集:软件运行时信息的获取依赖于监控探针的实现和织入^[13],包括运行时软件本身及其环境的信息。

2)监控信息分析诊断:对软件在运行阶段产生的数据进行分析 and 挖掘以评估系统运行状态和诊断故障,据此对软件系统进行动态调整和问题反馈,为软件在线演化提供数据来源与支撑。

(5)在线演化服务

在线演化是指为了提高软件系统适应需求的能力,在不中断其所提供服务的前提下,自动或借助外部动态指导发生的软件改变活动^[14],目前有基于组件的ACE(Adaptive Communication Environment)演化机制、基于函数/对象/类的FOEM(Flexible Online Evolution Mechanism)演化机制^[15],以及基于体系结构的演化机制^[16,17]等。在线演化使能平台模型有OSGi框架、Fractal平台、PKUAS平台、ArchStudio平台等^[14],演化机制不同,平台结构存在差异。

5.1.2 平台主要特征

一个值得信赖的、优秀的软件开发、共享与演化平台应当至少具备以下特征:

(1)简单性。具有利于理解的编码格式和使用方式,便于设计、开发、维护和使用。

(2)开放标准。基于一套完整、通用且开放的标准实现。

(3)灵活性。具有非常灵活的集成性,为应用提供松散耦合的服务方式。

(4)跨平台性。不限应用开发语言和运行平台,只要需要均可调用。

(5)安全性。提供统一的安全认证与保护服务。

腾讯开放平台、苹果公司的AppStore^[18]、IBM公司的Rational Jazz^[8]以及由国防科学技术大学等多家院校联合研制的Trustie1.0^[8]都是软件开发、共享与演化平台的典型实践。以腾讯为例,腾讯公司在2011年发布了腾讯开放平台,截止2015年,腾讯开放平台的开发者已达500万,应用已达240万款,涵盖了娱乐、生活、教育等内容,不仅为用户带来了便利,更为开放平台的合作伙伴带来了巨大的收益^[19]。

建立共享平台可以实现软件组件标准化与集成组装,实现面向用户的统一调用。开发行为可统计、可评估、可干涉调整,实现开发过程的可控竞争,从而提升软件开发过程质量。

5.2 建立需求生成与演化机制

需求是软件系统演化过程的起点和关键点,个性化和多元化特点决定了其演化的复杂性。群体智能模型下,以群体创作的思维开展需求获取活动,需求获取对象不再局限于系统用户,还包括开发者、项目相关者等。对于软件的需求变更管理问题,国内外的软件业研究人员从学术到实践给出了各

种解决方案,包括 CMM、ISO9000、SPICE、RUP 和 XP 需求管理软件等,这些方法有的过于笼统,有的缺乏组织管理和制度,有的又过于具体,可操作性差。现实中的需求管理大多是不成熟、不规范的^[20],但是建立需求管理框架、定义软件需求变更管理过程是要遵循的基本原则。

需求管理框架包括对需求变更的审核、管理与统计等,需求管理过程则至少应包括需求描述规范、需求演化模型和冲突消解机制。

(1)需求描述规范。即按照规则形式化地描述需求,是站在用户的角度对需求进行精确表达,可以采用自然语言、形式化和半形式化的语言描述。

(2)需求演化模型。即以一种可控、有序的方式完成需求演化,需求变更确认前通常要经过需求演化请求、演化过程控制、演化约束检查等过程,以确保需求合理、正确的演化。

(3)需求冲突消解。当多个需求演化请求同时提出时,可能存在内部和外部的冲突,必须对潜在冲突(互逆、覆盖、触发、相似)进行检测、分析和消解。

需求生成与反馈制度的建立,有益于用户满意度的提升,能够直接改善用户体验,带来软件质量提升的直观感受。

5.3 基于数据反馈的质量持续改进

基于数据的精细化项目管理能够减少成本,实时监控项目中软件代码质量的变化,快速定位导致软件退化的原因,对项目管理人员具有重大意义^[20]。这一方法逐渐在业界得到认可,同样适用于开发团队较多的复杂软件系统,其中,数据采集的有效性 & 改进模型的合理性对基于数据的系统质量改进效果至关重要。建立如图 2 所示的质量改进模型,在各个环节分别实施数据采集与分析挖掘。

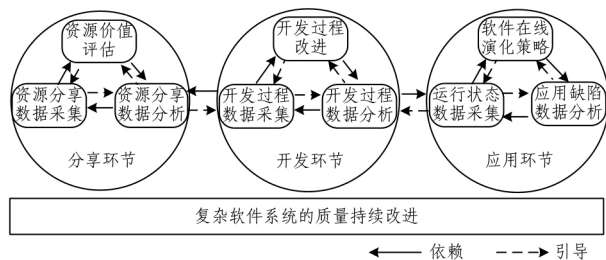


图 2 基于数据的质量改进模型

(1)开发环节。开发过程数据包括版本库、提交日志、缺陷库、阶段产品等,基于开发过程数据可以分析软件开发人员的力量投入和研制水平,预测软件缺陷。

(2)分享环节。资源分享数据包括查看下载次数、更新频率、用户评价及反馈意见等,基于分享数据可以分析软件需求度、功能缺陷、用户愿景等。

(3)应用环节。应用运行监控数据包括各类应用环境下软件实例的行为数据和环境反馈数据,基于运行监控数据可以快速定位故障、发现系统潜在威胁、均衡系统资源分配、决策在线演化策略等。

3 个环节产生的数据与分析结果可以相互利用、转化,实现软件开发过程的量化监控与评估,快速、直接反馈作用于软件开发过程,是对软件的环境适应性、健壮性和可靠性的直接提升。

结束语 近年来,群体软件工程方法逐渐受到关注,它强调群体性、大众化的软件研发竞争机制对软件资源产生的促进作用,也包括计划性、精英化的软件整体规划和管理对软件架构、组织法则、社区规范的形成具有的宏观决策作用,面向

解决基于云计算与物联网的超大规模信息处理、信息时刻变化且散落在世界各地的问题。本文所提的基于群集智能的复杂软件系统开发模型仍主要用于组织性较强的软件开发过程,强调群集智能在开发过程中的应用,目的在于构建相对隔离的复杂软件系统生长与演化的生态环境,解决局部系统间的连接问题,促进组织内部的协作开发与资源共享,激活技术创新。与群体软件工程已迈入社会工程领域有所不同,本文所提模型仍在机器工程范畴。

本文提出的复杂软件系统开发过程改进方法是从部分项目成功实践中总结提取,还需要在更广泛的应用中细化与优化,考虑上升为复杂软件系统开发过程规范,配套以管理制度,从而指导后续复杂软件系统的研制。

参考文献

- [1] 汤伟君,邱美虹.复杂系统、突现及其对科学教育的启示[J].科学教育月刊,2007,301:17-24
- [2] 王怀民,吴文峻,毛新军,等.复杂软件系统的成长性构造与适应性演化[J].中国科学:信息科学,2014,44:743-761
- [3] 姜茸,杨明.软件复杂性研究综述[J].计算机系统应用,2014,23(9):1-5
- [4] 吴大川.软件开发过程管理研究[D].大连:大连理工大学,2002
- [5] 肖人彬,等.面向复杂系统的群集智能[M].北京:科学出版社,2013:20-25,211-214
- [6] 张卫华.基于 web 服务的在线演化技术研究[D].广州:广东工业大学,2013
- [7] 肖人彬,陶振武.群集智能研究进展[J].管理科学学报,2007,10(3):80-96
- [8] 王怀民,尹刚,谢冰,等.基于网络的可信软件大规模协同开发与演化[J].中国科学:信息科学,2014,44(1):1-19
- [9] Team T. Trustie Collaborative Development Environment Reference Specification[R]. Jinan: TRUSTIE-FORGE V2.0. 2011
- [10] 袁霖,王怀民,尹刚,等.开源环境下开发人员行为特征挖掘与分析[J].计算机学报,2010,33:1909-1918
- [11] 周明辉,郭长国.基于大数据的软件工程新思维[J].中国计算机学会通讯,2014,10(3):37-41
- [12] 蔡斯博,邹艳珍,邵凌霄,等.一种支持软件可信评估的框架[J].软件学报,2010,21(2):359-372
- [13] 刘东红.大规模分布式软件系统的伴随式监控技术研究[D].长沙:国防科学技术大学,2011
- [14] 王怀民,史佩昌,丁博,等.软件服务的在线演化[J].计算机学报,2011,34(2):318-328
- [15] 陈洪龙,李仁发.一种灵活的软件在线演化机制[J].计算机科学,2010,37(5):115-117
- [16] Peyman O, Nenad M, Taylor Richard N. Architecture-based runtime software evolution[C]// Proceedings of the International Conference on Software Engineering, 1998 (ICSE'98). Kyoto, 1998:177-186
- [17] Peyman O, Gorlick Michael M, Taylor Richard N, et al. An architecture-based approach to self-adaptive software[J]. IEEE Intelligent Systems, 1999, 14(3):54-62
- [18] 徐晓兰.一种群体软件工程过程开发方法:中华人民共和国, CN 104267973 A[P].北京万方数据股份有限公司,2015
- [19] 马化腾.互联网+:国家战略行动路线图[M].北京:中信出版集团,2015:141-159
- [20] 赵晶华,苏秦,等.软件需求变更过程质量管理及控制初探[J].计算机应用研究,2004,7:31-34