

一种新的软件开发协作模型研究

司艳丽¹, 刘万辉²

(1 淮阴师范学院 计算机科学与技术学院, 江苏 淮安 223300)

2 淮安信息职业技术学院 计算机与通信工程学院, 江苏 淮安 223003)

摘 要: 通过活动理论研究, 提出了一种新的协同软件开发模型, 文中对该模型框架的各个组成部分进行了详细的论述, 并对该模型进行了形式化定义和描述. 该模型适于大型软件的开发过程, 可以使开发过程更加协同, 并提高软件开发的质量.

关键词: 协同; 活动理论; 大型软件开发

中图分类号: TP301

文献标识码: A

文章编号: 1000-7180(2012)05-0073-04

Research of a New Software Development Cooperative Model

SI Yan-li¹, LIU Wan-hui²

(1 School of Computer Science and Technology, Huaiyin Normal University, Huaian 223300, China;

2 School of Computer Science and Communication Electronic, Huaian Vocational College of Information, Huaian 223003, China)

Abstract: By studying the activity theory, a new software development cooperative model was presented in this paper. The paper described the components and gave a formal definition for the framework model. The model can apply to the process of largesized software development, it make the developing process more cooperation and improves quality of software.

Key words: cooperation; activity theory; largesized software development

1 引言

规模较大的软件开发过程本质上是一种分布式协同工作过程, 其主要通过多媒体、Internet 等技术, 将工作上需互相协作的多个成员或开发团队有机的组织起来, 完成软件开发任务^[1-2]. 这一过程本质上属于计算机支持协同工作 (Computer Supported Cooperative Work, CSCW)^[3].

在传统的软件开发过程中, 软件开发是按顺序是从概要设计到详细设计, 分模块完成的, 前一模块全部完成后, 再进行下一模块工作. 因此这一过程基本上属于分散、静态、顺序的流程. 其结果往往使得软件开发周期延长, 效率不高、过程的适时反馈性能低下, 也没有正确的计划原理支持. 因此, 一种深层次的过程管理显得越来越重要. 但是, 目前的理论与

技术还不能适用于大规模软件开发下的协同工作. 因此, 研究一种新的软件开发协作模型就显示出其必要性.

本文提出了一种基于活动理论 (Activity Theory, AT) 的协作软件开发模型, 并对其进行了形式化描述. 期待该模型应用于大型软件开发过程中可以提高软件开发质量, 缩短开发周期, 使开发过程更加协同.

2 活动理论

AT 最早由 Vygotsky 等人最早提出^[4], 其主要是在心理学研究领域, 用于分析和建模人类的活动. AT 的基本思想是: 将人类的某一活动作为一个个体研究单位, 这些活动通过工具建立联系、互相协调进行. Vygotsky 的 AT 观点可由图 1 所示.

收稿日期: 2011-09-12; 修回日期: 2011-10-10

基金项目: 国家自然科学基金项目 (70671025); 江苏省自然科学基金项目 (SBK200921319)

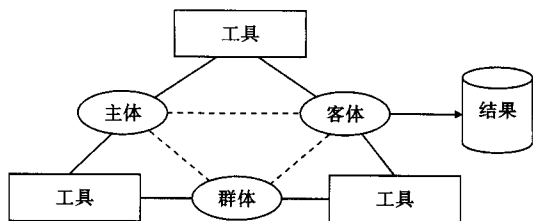


图1 AT的层次结构

AT思想可通过三元组表示(Subject-Tool-Object)^[3]. Engeström在AT思想的基础上提出了更加完善的活动框架,其主要贡献是:在原有的AT思想中引入了文化因素和社会因素^[5],如图2所示。

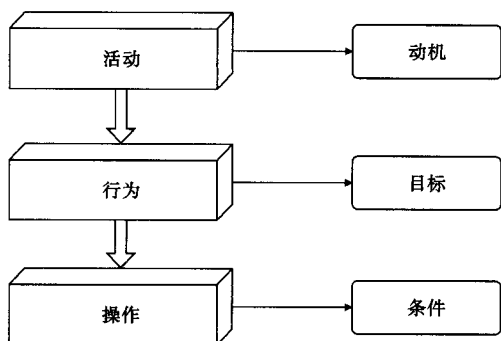


图2 活动理论框架

在AT中,活动的组成要素包括:主体(subject)、客体(object)、工具(tool)、规则(rule)、共同体(communitiy)和劳动分工(division of labour)^[6-7]. AT思想中所强调的主体作用于客体的活动是在工具(tool)、规则(rule)、共同体(communitiy)和劳动分工(division of labour)共同参与下进行的. 因此,完善后的AT将活动分为“活动”、“行为”和“操作”三个层次(如图2). AT的相关概念如下:

(1) AT分层结构:在AT中,分解后的每一个个体活动单元的行为必须是为了满足活动所指的对象,即每一个个体活动单元都有明确的指向. 行为具有意识性,但不同行为的最终目标是一致的. 在分层结构内部活动可以根据需要进行适时的控制,并可以随着约束条件的变化而改变.

(2) AT的面向对象性:我们生活在一个客观的世界中,客观存在的事物同时具有自然科学的客观存在性和社会和文化所定义的某些其他特性,任何活动都具有目标性,并指向它的目标活动(对象).

(3) AT内转化:AT中,活动可分为内部和外部两种,两者可以相互转化. 内转化就是外部活动向

内部活动转化. 内转化主要作用在于将潜在的事物转化为实际事物,其过程不需要实际操作参与.

(4) AT外转化:外转化是将内部活动向外部活动转化. 一般情况下,当一个内部的活动面临修正或调整时,外转化就显得十分必要.

(5) 调解工具:AT强调人类的活动主要通过工具进行调解. 工具是在活动的发展过程中创造和改进的,其携带了相关的历史变化信息. 因此,工具可以看作是社会知识的传递与积累.

(6) AT的开发方法:这里所指的开发,不仅是针对某一个具体的对象,其还包含了一般的方法学. AT中主要强调造型试验(formative experiment),而非传统意义上的通过实验室进行的试验. 造型试验要求进行历史追踪,同时注重项目管理者实践开发.

3 基于AT思想的软件开发协作模型

软件开发活动将目标转换成一个软件制品或软件产品. 软件开发活动是由一个团队协作来完成,有一个共同的目标. 软件开发活动由一系列行为组成,每个行为有特定的任务. 每个行为又由一系列操作完成,每个操作是在特定的条件下执行^[5]. 操作可以看作是活动的最小单位,可以理解为具有原子性. 操作的执行需要特定的工具. 软件过程是软件工程领域一个重要研究问题. 一个软件开发过程可以由一组软件活动组成,活动之间可以串行的,也可以是并行的,活动也可以反复迭代进行,从而成为一个动态的活动网络.

根据软件开发工作自身的特殊性与层次结构,文中在AT思想的基础上针对软件开发工作过程提出了以活动、行为、操作为组成元素的三层架构软件开发模型. 该模型中将软件开发工作过程分为活动层、行为层和操作层. 利用活动模型、会话、会议进行对该三层架构进行描述. 如图3所示.

3.1 活动层描述

软件开发的整个工作过程可以被认为是一组各自独立而又彼此关联的活动,该组活动同时具有串行性与并行性. 软件开发过程针对不同种类的开发任务存在个体差异,但所有软件开发过程都存在一定的共性. 较好的情况是某个软件开发过程可以被视作一组粗粒度工作流,这样其整个软件开发过程的并行性较好. 但就软件开发过程自身来讲,必定存在某些环节可实现并行.

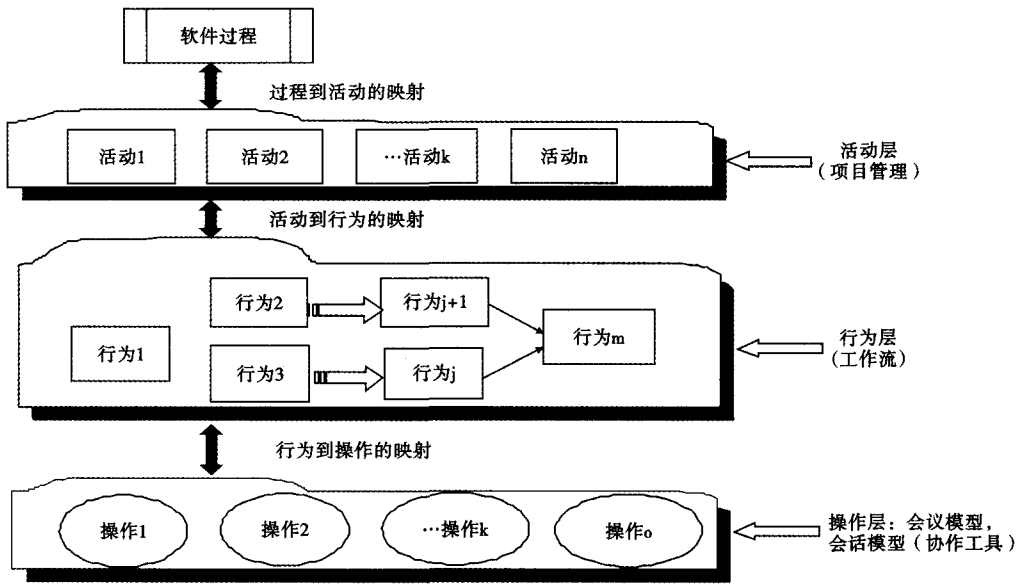


图3 基于活动理论的软件开发协作模型

3.2 行为层描述

深入到具体活动的内部,将具体的活动进行二次分解.即,将某一具体活动分解为一组各自独立而又彼此关联的行为,该组行为同时具有串行性与并行性.也可以理解为在活动内部形成一组由行为构成的工作流.这其中涉及到了具体的角色和资源,因此可以用AT模型对其进行描述和实现.

3.3 操作层描述

如果将行为再进行分解,则行为由具体的操作构成.同样操作之间的彼此关系可以是并行或者串行.类似于数据库理论,在这里操作可以分为协同,并发和独占操作.其中,如何实现并发和协同将是这一层主要解决的问题.为了对操作层进行描述,文中引入了“会话模型”、“会议模型”等.

4 形式化定义

在上节软件开发协作模型的基础上,本节对基于活动理论的协作软件开发模型进行了形式化描述.为了描述方便给出如下符号:

- (1) R_o :角色元素集合.
- (2) R_t :工件元素集合.
- (3) T :工具元素集合.
- (4) G :目标元素集合.
- (5) U :成员元素集合.
- (6) $Element$:基本元素集合,由以上元素构成,即: $Element = \{R_o, R_t, T, G, U\}$
- (7) r :关系集合.

(8) ϕ :AT中所指的个体操作; δ :AT中所指的个体行为; λ :AT中所指的个体活动.

(9) Grp :软件开发实施群体.

(10) Prc :协同软件开发过程.

4.1 对操作(ϕ)、行为(δ)与活动(λ)等基本概念的形式化描述

(1) 如上所述操作 ϕ 可形式化表示如下:

$\phi = \{R_o, R_t, T, f_\phi\}$,其中 f_ϕ 为关于操作 ϕ 的映射,描述了工具元素与工件元素的对应关系,可形式化表示为 $f_\phi: R_t \rightarrow 2^T$.

假设 ro 为 R_o 中的一个角色 $ro \in R_o$,可表示如下:

$\phi_{ro} = \{ro, R_t, T, f_\phi\}$,其中 $\phi_{ro} \in \phi$.该描述使得角色 ro 使用特定的工具 T 在工件 R_t 上完成相关操作 ϕ .

(2) δ 可形式化表示如下:

$\delta = \{R_o, \phi, f_\delta\}$,其中 f_δ 描述了个体行为与操作之间的映射关系,可形式化表示为 $f_\delta: \delta \rightarrow 2^\phi$.

同上,假设 ro 为 R_o 中的一个角色,可表示如下:

$\delta_{ro} = \{ro, \phi, f_\delta\}$,其中 $\delta_{ro} \in \delta$.以上描述了角色 ro ,通过一组操作 ϕ ,完成了特定的行为 δ .

(3) λ 可形式化表述如下:

$\lambda = \{R_o, \delta, f_\lambda\}$,其中 f_λ 描述了个体行为与个体活动之间的映射关系,可形式化表示为 $f_\lambda: \lambda \rightarrow 2^\delta$.

同上,假设 ro 为 R_o 中的一个角色,可表示如下:

$\lambda_{ro} = \{ro, \delta, f_{\delta}\}$, 其中 $\lambda_{ro} \in \lambda$. 以上描述说明个体活动 λ 由角色 ro 承担, 并且可根据个体活动与个体行为之间的映射关系, 将 λ 分解成若干协同行为 δ .

4.2 对各种关系的形式化描述

(1) 个体行为之间关系 (Action Relation, AR) 描述如下:

$AR = \{\delta, f_{AR}\}$, 其中映射 f_{AR} 可形式化描述为: $f_{AR}: \delta \times \delta \rightarrow \{\text{true}, \text{false}\}$ (“ $\times$ ”为笛卡尔积运算符). 以上描述含义为个体活动中不同行为之间的继承关系, 如果继承关系存在则 f_{AR} 为真(true), 否则为假(false).

(2) 个体活动之间关系 (Active Relation, ActR) 描述如下:

$ActR = \{\lambda, f_{ActR}\}$, 其中映射 f_{ActR} 可形式化描述为: $f_{ActR}: \lambda \times \lambda \rightarrow \{\text{true}, \text{false}\}$. 以上形式化同样描述了不同个体活动之间的继承关系, 如果继承关系存在则 f_{ActR} 为真(true), 否则为假(false).

(3) 访问关系 (Access Relation, AccR) 描述如下:

$AccR = \{Ro, \phi, Rt, f_{AccR}\}$, 其中, $f_{AccR}: Ro \times T \rightarrow 2^{\phi}$, 映射 f_{AccR} 描述了角色 Ro 在特定的时间 t , 对工件 Rt 所能作的操作 ϕ .

(4) 角色关系 (Role relation, Rr) 描述如下:

$Rr = \{U, Ro, f_{Rr}\}$, 其中, $f_{Rr}: U \times t \rightarrow 2^T$, 映射 f_{Rr} 描述了成员 U 在特定时间 t 所扮演的角色 Ro .

4.3 软件过程 (Process, P) 到活动的关系的形式化描述

(1) 首先, 给出群体 Grp 的形式化描述:

$Grp = \{U, f_{Grp}\}$, 其中, $f_{Grp}: Grp \rightarrow 2^U$.

(2) 软件过程 P 的形式化描述

$P = \{Grp, G, \lambda, f_P\}$, 其中, $f_P: G \rightarrow 2^{\lambda}$. f_P 描述了软件开发过程到个体活动的映射关系. 据此, 协作开发过程也可以定义为群体 Grp 为共同的目标 G 所执行的一组活动 λ .

综上, 可以给出所有关系 (Relation, R) 组成的集合: $R = \{f_{\phi}, f_{\delta}, f_{\lambda}, f_{AR}, f_{ActR}, f_{AccR}, f_{Rr}, f_{Grp}, f_P\}$. 因

此, 基于 AT 思想的协作软件开发模型可以被描述为 $\langle \text{Element}, P, R \rangle$ 三元组.

5 结束语

本文提出了一种新的协作软件开发模型. 由于 AT 在描述软件开发领域具有其自身的优势^[3], 因此, 期待本文提出的模型会在理论层面对软件协同开发领域研究起到促进作用. 目前为止, 该模型的实际应用还有待进一步研究, 其高效性还有待进一步验证.

参考文献:

- [1] 章宇, 葛宁, 冯重熙. 使用软硬件协同仿真技术的嵌入式系统开发工具包[J]. 微电子学与计算机, 2006, 23 (7): 43-46.
- [2] 陈桦, 陈景霞. 基于 XML 的机床产品协同设计信息交互模型的研究[J]. 微电子学与计算机, 2005, 22 (2): 164-166.
- [3] 史美林, 向勇, 杨光信. 计算机支持的协同工作理论与应用[M]. 北京: 电子工业出版社, 2000: 16-22.
- [4] Nardi B A. Activity theory and human—computer interaction[M]. Cambridge: MIT Press, 1996: 5-22.
- [5] Engeström. Activity theory and individual and social transformation [P]. Lahti, Finland, Congress for Reaserach on Activity Theory. 1990, 21-25.
- [6] Kuutti K. The concept of activity as a basic unit of analysis for CSCW research[C] // ECSCW' 91. UK: London, 1991: 249-264.
- [7] Kuutti K. Identifying potential CSCW application by means of activity theory concepts: a case example[C] // Proc. of CSCW' 92. Canada: Toronto, 1992: 233-240.

作者简介:

司艳丽 女, (1979—), 硕士, 讲师. 研究方向为数字图像处理技术、软件工程等.

刘万辉 男, (1978—), 硕士, 讲师. 研究方向为软件工程等.